RINKTINIAI KOMBINATORIKOS SKYRIAI, I-II

Eugenijus Manstavičius
2017

Pirmoji dalis parengta pagal Th.H. Carmen, Ch.E. Leiserson ir R.L. Rivest knygos *INTRODUCTION TO ALGORITHMS* , (MIT Press, Mc Graw-Hill, 1990) VI skyrių, antroji - pagal R. Sedgewick ir P. Flajolet, *ANALYTIC COMBINATORICS* (Cambridge University Press, 2007). Žemiau talpinama medžiaga yra suklijuota iš autoriaus ilgesnių pasirenkamųjų kursų, todėl dėl techninių sunkumų teko atsisakyti vieningos puslapių numeracijos.

Contents

Pratarmė

Šis konspektas nėra pažodinis nurodyto knygos skyriaus vertimas. Pagrindinis dalykas, kuris yra pasisklolintas iš šios knygos – medžiagos parinkimas, jos temų išdėstymo eiliškumas. Sunku būtų čia rasti tobulesnį ir originalesnį būdą. Mes stengėmės įpiršti savąjį kelią į grafų teorijos algoritmų supratimą bei jų analizę. Studijuodamas kiekvieną temą, Skaitytojas turėtų pasiruošti grafinius algoritmų veikimo eskizus, jei tokie brėžiniai neįdėti. Grafų vizualizacijos medžiaga bus paruošta artimiausioje ateityje.

1. Grafų reprezentacija

Grafą (multigrafą, digrafą, multidigrafą) žymėsime $G = (V, E)$, čia V yra viršūnių aibė, o E – briaunų aibė (multigrafų atveju – multiaibė). Jų galios $|V| = n$ ir $|E| = m$ vadinamos grafo eile ir jo didumu. Dažniausiai informacija apie grafus užrašoma sudarant gretimumo sąrašus $Adj[u]$. Tai yra viršūnės $u \in V$ gretimųjų viršūnių sąrašas (digrafų atveju – viršūnių, į kurias veda lankai iš u, sąrašas). Informacija gali būti pateikiama ir matriciniais būdais.

Tuo tikslu reikia skaičiais $\{1, 2, ..., n\}$ sunumeruoti grafo viršūnes. Pažymėkime a_{ij} kiekį briaunų, jungiančių i-ąją ir j-ąją viršūnes multigrafe G, kai $i \neq j$, ir a_{ii} - dvigubą kilpų, išvestų iš i-os viršūnės, skaičių. Kvadratinė matrica $A = ((a_{ij}))$, $1 \leq i, j \leq n$, vadiname multigrafo *gretimumo matrica.* Jei multigrafas neturi kilpų, tai matricos pagrindinėje įstrižainėje yra nuliai. Multigrafo gretimumo matrica yra simetrinė.

Norėdami užfiksuoti, kurios briaunos jungia kurias viršūnes, įvedame dar vieną matricą. Dabar skaičiais $\{1, 2, ..., m\}$ sunumeruojame ir briaunas. Multigrafų atveju, žinoma, numeruojamos visos briaunos bei kilpos.

Matricą $B_G = B = (b_{ij})$, $1 \leq i \leq n$, $1 \leq j \leq m$, vadiname *multigrafo incidentumo matrica*, jei

$$b_{ij} = \begin{cases} 1, & \text{jei } i \text{ viršūnė yra incidenti } j \text{ briaunai, kuri nėra kilpa,} \\ 2, & \text{jei } i \text{ viršūnė yra incidenti } j \text{ briaunai, kuri yra kilpa,} \\ 0, & \text{jei } i \text{ viršūnė nėra incidenti } j \text{ briaunai.} \end{cases}$$

Apibrėžiant *multidigrafo* gretimumo bei incidentumo matricas, atsižvelgiama į briaunos (lanko) kryptį. Gretimumo matricos elementai a_{ij} lygūs skaičiui briaunų, išvestų iš i-tos į j-ą viršūnę. Kilpos atveju skaičius nebedvigubinamas. Digrafo gretimumo matrica nebūtinai simetrinė.

Bekilpiam multidigrafui incidentumo matricos elementai

$$b_{ij} = \begin{cases} 1, & \text{jei } i \text{ yra pradinė } j \text{ briaunos viršūnė,} \\ -1, & \text{jei } i \text{ yra galinė } j \text{ briaunos viršūnė,} \\ 0, & i \text{ viršūnė nėra incidenti } j \text{ briaunai.} \end{cases}$$

Jei i viršūnė yra incidenti kilpai, pažymėtai j numeriu, tai dažnai vartojamas žymuo $b_{ij} = -0$.

Pateiksime vieną įvestųjų matricų sąryšį, rodantį, kad ne visos matricos gali būti gretimumo ir incidentumo matricomis.

Teorema. *Tarkime, G yra numeruotas multidigrafas be kilpų, A ir B – jo gretimumo ir incidentumo matricos atitinkamai. Tada*

$$BB' = D - A.$$

Čia $'$ žymi matricos transponavimą, o D – diagonali matrica, kurios įstrižainėje yra iš eilės surašyti viršūnių laipsniai.

Įrodymas. Jei c_{ij} – matricos BB' bendrasis narys, tai

$$c_{ij} = \sum_{l=1}^{m} b_{il} b_{jl}. \tag{1}$$

Todėl, kai $i \neq j$, sandauga $b_{il}b_{jl}$ lygi 0 arba -1. Pastaroji lygybė yra teisinga tik tuo atveju, kai $x_i x_j = e_l$. Sudedant pagal l, -1 dauginsis iš tokio skaičiaus, kiek yra briaunų, jungiančių x_i ir x_j.

Kai $i = j$, c_{ii} yra matricos įstrižainės narys. Matrica A turi nulinę įstrižainę. Dabar (1) suma lygi briaunų, išvestų iš x_i skaičiui.

Teorema įrodyta. ◇

2. Paieška į plotį

Užduotis: Pradėjus fiksuota viršūnės s reikia atrasti (pereiti) visas pasiekiamas grafo (digrafo) viršūnes keliaujant jo briaunomis (digrafe – jos kryptimi).

Tako $u - v$ ilgiu vadinamas pereinamų briaunų grafe skaičius. *Atstumu* tarp u ir v vadinamas minimalus tako ilgis. Žymėsime $\delta(u, v)$. Susitariama, kad $\delta(u, v) = \infty$, jei u nepasiekiama iš v. Jei x, u, y – trys gretimos viršūnės, esančios take, tai x vadinsime viršūnės u *tėvu*, o y – jos *vaiku*.

Paieškos į plotį algoritmas atlieka nurodytą užduotį. Jį realizuojant, informacija pateikiama grafo gretimumo sąrašu. Vykdant algoritmą viršūnėms priskiriami skaitinis $d[u]$, spalvos $c[u]$ bei tėvystės $\pi[u]$ atributai. Taip pat formuojama apdorojamų viršūnių eilė Q. Jos pirmąjį elementą žymėsim $h[Q]$ ir vadinsim *galva*. Naujos viršūnės v prirašymas šios eilės gale bus operacija $ENQUEUE(Q, v)$, o galvos išėmimas iš sąrašo – operacija $DEQUEUE(Q)$. Kai atributas višūnei nepriskiriamas jį prilyginsime *NIL*-ui.

Simboliniais kodais paieškos į plotį algoritmą galima užrašyti taip:

P-Plot(G, s)**:**

1 **for each** $u \in V - \{s\}$

2 **do** $c[u] \leftarrow$ *balta*

3 $d[u] \leftarrow \infty$

4 $\pi[u] \leftarrow NIL$

5 $c[s] \leftarrow$ *pilka*

6 $d[s] \leftarrow 0$

7 $\pi[s] \leftarrow NIL$

8 $Q \leftarrow \{s\}$

9 **while** $Q \neq \emptyset$

10 **do** $u \leftarrow h(Q)$

11 **for each** $v \in Adj[u]$

12 **do if** $c[v] =$ *balta*

13 **then** $c[v] \leftarrow$ *pilka*

14 $d[v] \leftarrow d[u] + 1$

15 $\pi[v] \leftarrow u$

16 *ENQUEUE*(Q, v)

17 *DEQUEUE*(Q)

18 $c[u] \leftarrow$ *juoda*

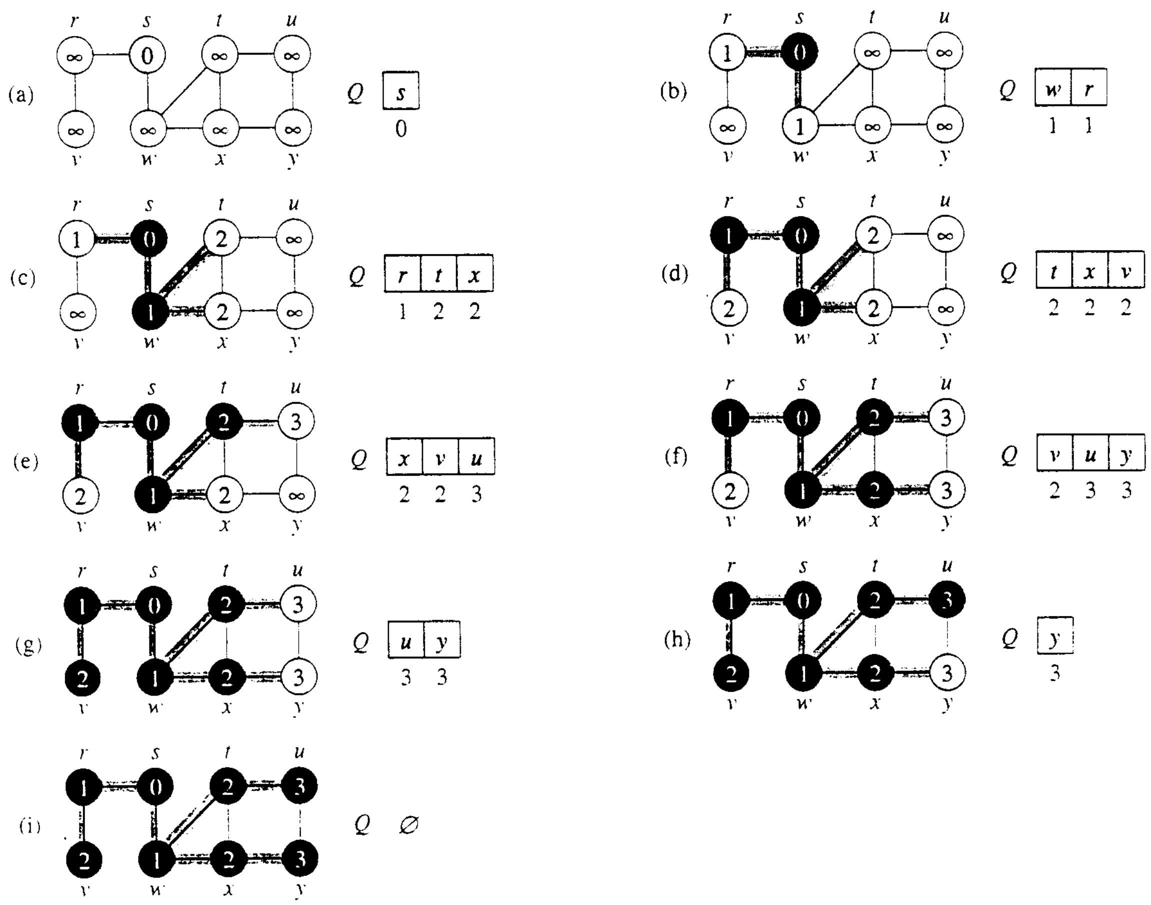
END

Taigi, 1-4 eilutės inicializuoja atributus, Q pradedama 8 eilutėje, ji visada tesiama pilkomis viršūnėmis. 9–18 eilutės – sudaro pagrindinę algoritmo dalį.

Įvertinkime procedūros *P-Plot*(G, s) vykdymo trukmę. Vienos viršūnės įjungimas į eilę Q procedūra *ENQUEUE*(Q, s) užtrunka $O(1)$ laiko, tiek pat – *DEQUEUE*(Q). Į eilę viršūnė patenka po 11 žingsnio ir tik baltos viršūnės. Tad, viršūnė patenka ne daugiau kaip vieną kartą. Išmetimas iš sąrašo, jei viršūnė ten buvo, vykdomas tik kartą. Vadinasi, šios procedūros trunka $O(|V|) = O(n)$ laiko vienetų.

Gretimumo sąrašas skaitomas tik vieną kartą. Jo eilučių ilgių suma lygi $O(|E|) = O(m)$. Taigi, algoritmo *P-Plot*(G, s) vykdymo laikas yra $O(|V| + |E|) = O(n + m)$.

Išnagrinėkime eskizą.



Ką randame įvykdę algoritmą? Vienos viršūnės nudažytos juodai, kitos baltai, turime atributų $d[v]$ baigtines arba begalines reikšmes. Yra užfiksuoti protėviai $\pi[v]$. Pasirodo, kad to pakanka nurodyti vieną iš trumpiausių $s - v$ takų, kurį žymėsime $T(s,v)$, nurodydami jame esančias briaunas. Griežtai kalbant, realizavus algoritmą mes randame dydžius, nurodytus šioje teoremoje.

1 teorema. *Įvykdžius algoritmą, turime*

1) kiekvienai $v \in V$

$$\delta(s,v) = d[v];$$

2) atributas $d[v] = \infty$*, kurį turi baltosios viršūnės, nurodo visas nepasiekiamas iš* s *viršūnes;*

3) kiekviena juoda viršūnė v *yra pasiekiama iš* s*, be to, vienas iš trumpiausiųjų takų* $T(s,v)$ *gali būti apibrėžtas rekurentiškai:* $T(s,s) = \{\emptyset\}$ *ir* $T(s,v) = T(s,\pi[v]) + \pi[v]v$.

Keletą reikalingų pastebėjimų pateiksime lemomis.

1 lema. *Jei* $(u, v) \in E$, *tai* $\delta(s, v) \leq \delta(s, u) + 1$. *Čia taip pat laikoma, jog* $\infty \leq \infty + 1$.

Įrodymas. Akivaizdu. $\diamond$

2 lema. *Įvykdžius algoritmą,* $\delta(s, v) \leq d[v]$ *kiekvienai* $v \in V$.

Įrodymas. Jei v nepateko į eilę Q, ji išlieka balta ir po inicializavimo $\delta[v] = \infty$. Tad lemos tvirtinimas šiuo atveju trivialus.

Patekusias į sąrašą Q viršūnes galime sunumeruoti pagal patekimo laiką ir šį numerį galime laikyti indukcijos parametru. Pirmajai viršūnei s turime $d[s] = 0$ ir $\delta(s, s) = 0$, tad pirmasis indukcijos žingsnis atliktas.

Tegu lemą jau įrodėme dėl ankstesnių viršūnių ir v yra sekanti mūsų numeracijoje viršūnė. Ji pateko į eilę, kai buvo nagrinėjamas, kažkokios jau pilkos viršūnės $u = h(Q)$, gretimumo sąrašas. Todėl $v \in Adj[u]$ ir pagal indukcijos prielaidą gauname, kad $d[u] \geq \delta(s, u)$. Eilutėje 14 priskyrėm $d[v] = d[u] + 1$. Vadinasi, $d[v] \geq \delta(s, u) + 1$. Pasinaudoję 1 lema, baigiame ir šį indukcijos žingsnį. $\diamond$

3 lema. *Kiekvienai eilei* $Q = \{v_1, \ldots, v_r\}$, *čia* $v_1 = h[Q]$ *yra jos galva, atsiradusiai vykdant algoritmą, turime* $d[v_i] \leq d[v_{i+1}]$, $1 \leq i \leq r - 1$, *ir* $d[v_r] \leq d[v_1] + 1$.

Įrodymas. Žodis "kiekvienai" tiesiog įpareigoja taikyti indukciją sąrašų Q atžvilgiu. Tad, juos reikia kažkaip sustatyti į eilę. Nauji sąrašai atsiranda vykdant *ENQUEUE*(Q, s) ir *DEQUEUE*(Q). Galime atskirai surikiuoti į eilę sąrašus Q vykdant šias procedūras.

Pirmuoju atveju, jei $Q = \{s\}$, tai $d[s] = 0$ ir tvirtinimas yra teisingas. Tegu jis teisingas eilei $Q = \{v_1, \ldots, v_r\}$ ir jos gale įrašom v_{r+1}. Pastebėkime, kad $v_{r+1} \in Adj[v_1]$. Tad, $d[v_{r+1}] = d[v_1] + 1$ ir pagal indukcinę prielaidą $d[v_r] \leq d[v_1] + 1$. Iš čia matome, kad $d[v_r] \leq d[v_{r+1}]$. Tvirtinimas šiai sąrašų daliai įrodytas.

Panašiai *DEQUEUE*(Q) metu eilė $Q = \{v_1, \ldots, v_r\}$ keičiasi į $Q = \{v_2, \ldots, v_r\}$. Monotoniškumas akivaizdžiai išlieka net ir tuščios eilės atveju. Be to, $d[v_r] \leq d[v_1] + 1 \leq d[v_2] + 1$.

Lema įrodyta. $\diamond$

Grįžtame prie 1 *teoremos įrodymo.* Jei v yra nepasiekiama iš s, tai $\delta(s, v) = \infty$. Tad, pasinaudoję 2 lema, matome, kad ši viršūnė negalėjo patekti į eilę Q, vadinasi, jos spalva yra balta, tėvas neapibrėžtas, o atributas $d[u] = \infty$ po inicializacijos išlieka nepakitęs.

Nagrinėkime pasiekiamas viršūnes. Joms $\delta(s, v) < \infty$. Suskirstykime jas į sluoksnius

$$V_k = \{v \in V : \ \delta(s, v) = k\}, \quad k = 0, 1, \ldots.$$

Reikia įrodyti, kad sluoksnio viršūnėms $d[v] = k$, kad jos yra juodos ir kad egzistuoja trumpiausias takas $T(s, v)$. Naudosimės indukcija k atžvilgiu. Kai $V_0 = \{s\}$, visi tvirtinimai akivaizdūs. Tegu jau juos įrodėm sluoksniui V_{k-1}. Tad, jei $u \in V_{k-1}$, turime $d[u] = k - 1$, ji yra juoda ir $T(s, u)$ apibrėžtas teoremoje nurodyta rekurenčiuoju sąryšiu.

Kadangi pagal 3 lemą priskiriami atributai $d[u]$ sudaro nemažėjančią seką, viršūnė v iš sluoksnio V_k negali būti atrasta anksčiau negu buvo išnagrinėtos visos prieš tai buvusių sluoksnių viršūnės. Iš tiesų, priešingu atveju gavusi atributą $d[v]$, neviršijantį $k - 1$, pagal 2 lemą turėtų būti arčiau nuo s nei per $k - 1$.

Aišku, kad $v \in V_k$ yra gretima kažkokiai viršūnei u iš V_{k-1}, t.y. $v \in Adj[u]$. Vadinasi, ji bus aptikta ir 14 žingsnyje priskirta $d[v] = d[u] + 1 = k - 1 + 1 = k$. Todėl ir $d[u] = \delta(s, u)$. Tuo metu u yra pilka ir yra eilės Q galva. Panašiai, 15-ame žingsnyje rastas tėvas $\pi[v] = u$. Todėl takas $T(s, u) + uv$ turės ilgį k. Kadangi ir atstumas iki s yra k, šis takas ir yra vienas iš trumpiausiųjų takų. $\diamond$

Pasiekiamos iš s viršūnės yra vienoje nagrinėjamo grafo komponentėje. Digrafo atveju nebūtinai stipriojoje komponentėje, kuri apibrėžiama, kaip pografis, kuriame iš bet kokios viršūnės einant briaunų kryptimis patenkama į bet kurią jo viršūnę. Realizavę algoritmą, galime apibrėžkime *protėvių pografį* $G_\pi = (V_\pi, E_\pi)$, imdami

$$V_\pi = \{v \in V : ; \pi[v] \neq NIL\} \cup \{s\}$$

ir

$$E_\pi = \big\{(\pi[v], v) \in E \colon\ v \in V_\pi - \{s\}\big\}.$$

2 teorema. *Protėvių pografis yra medis, kuriame kiekviena viršūnė yra pasiekiama iš s vieninteliu trumpiausiuoju keliu.*

Įrodymas. Kaip pastebėjome įrodydami 1 teoremą, tėvai yra priskiriami visoms pasiekiamoms iš s viršūnėms, todėl protėvių pografis yra jungus. Jo didumas $|E_\pi| = |V_\pi| - 1$, todėl jis yra medis. Medyje dvi viršūnes jungia tik vienas takas. Lieka įžvelgti, kad šis takas sutampa su 1 teoremoje nurodytu trumpiausiuoju taku. $\diamond$

Šioje teoremoje apibrėžtas medis vadinamas *paieškos į plotį* medžiu.

UŽDUOTIS. Sudarykite programą, spausdinančią paieškos į plotį medį.

3. Paieška į gylį

Tikslas – atrasti visas grafo (digrafo) viršūnes, vadovaujantis strategija: sekančią viršūnę atrasti einant briauna, išvesta iš ką tik atrastos viršūnės. Jei tokių briaunų nėra, grįžti vienu žingsniu atgal ir vėl vadovautis minėta strategija. Jei jau grįžome į pačią pirmą viršūnę, ir nebeturime neatrastų jai gretimų, tai peršokame į dar neatrastą grafo viršūnę ir tęsiame procesą. Taip yra suformuojmas *jungiantysis grafo miškas*. Paieškos į plotį algoritme mes gavome tik jungiantįjį vienos grafo komponentės medį.

Kaip ir paieškoje į plotį, algoritmo realizacijai reikalingas grafo gretimumo sąrašas. Procedūroje viršūnei $u \in V$ priskiriami skaitiniai laiko parametrai $d[u]$ ir $f[u]$, išreiškiantys jos apdorojimo pradžios ir pabaigos laikus spalvos $c[u]$ (balta, pilka ir juoda) bei tėvystės $\pi[u]$ atributai. Kai tėvystės atributas viršūnei nepriskiriamas jį prilyginsime *NIL*-ui. Taip pat yra sekamas globalusis laikas, žymimas "*laikas*". Simboliniais kodais paieškos į gylį algoritmą galima užrašyti taip:

P-Gyl(G):
1 **for each** $u \in V$
2 **do** $c[u] \leftarrow$ *balta*
3 $\pi[u] \leftarrow NIL$
4 *laikas* $\leftarrow 0$
5 **for each** $u \in V$
6 **do if** $c[u]$=*balta*
7 **then** *VISIT*(u)

Čia *VISIT*(u) žymi procedūrą:

VISIT(u):
1 $c[u] \leftarrow$ *pilka*
2 $d[u] \leftarrow$ *laikas* $\leftarrow$ *laikas*+1
3 **for each** $v \in Adj[u]$ (eik briauna uv)
4 **do if** $c[v]$=*balta*
5 **then** $\pi[v] \leftarrow u$
6 *VISIT*(v)
7 $c[u] \leftarrow$ *juoda*
8 $f[u] \leftarrow$ *laikas* $\leftarrow$ *laikas*+1
END

Taigi, 1-4 eilutėse yra inicializacijos komandos, o 7 ir 8-oji aprašo programos vykdymą. Viršūnė u bus pilka visame absoliutaus laiko intervale $[d[u], f[u]]$ ir tik jame. Kai ją nuspalviname juodai, 8 eilutėje pastumiame absoliutųjį laiką vienetu ir pradedame kito medžio formavimą. Taip reaizuojant programą yra randamas jungiantysis miškas:

$$G_\pi = (V_\pi, E_\pi), \qquad V_\pi = V, \quad E_\pi = \{\pi[v]v\colon\ v \in V, \pi[v] \neq NIL\}.$$

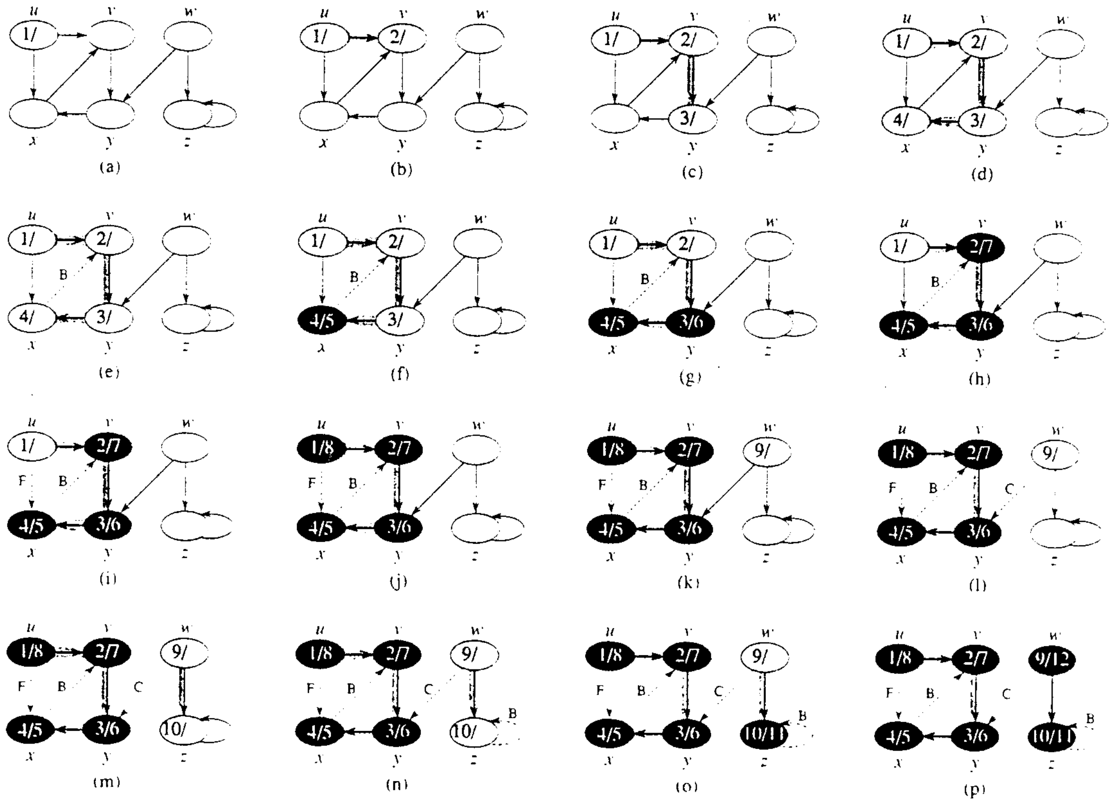
Pastebėkim, kad tėvą turi tos viršūnės u, kurioms buvo kviestas *VISIT*(u).

Aišku, kad 1-3 komandų vykdymas trunka $O(|V|)$ laiko vienetų. Panašiai, ir 5-6. Eilutėje 7 *VISIT* iškviečiama ne daugiau kaip $|V|$ kartų. Kaip rodo jos 3-6 eilutės, ši procedūra yra vykdoma kiekvienai sąrašo $Adj[u]$ viršūnei, tad iš viso ne daugiau kaip

$$\sum_{u \in V} |Adj[u]| = O(|E|)$$

laiko vienetų. Vadinasi, *P-Gyl*(G) vykdymo trukmė yra $O(n + m)$.

Algoritmo vykdymo eiga atspindėta brėžinyje:



Algoritmas akivaizdžiai atranda visas viršūnes, nes "peršoka" prie viršūnių, nepatekusių į ankstesnius medžius. Korektiškumas yra akivaizdus. Įsirodysime keletą paprastesnių teorinių šio algoritmo savybių.

1 lema (skliaustų l.) *Vykdant P-Gyl(G) grafe G kiekvienai porai $u, v \in V$ turime vieną iš trijų galimybių:*

(i) *intervalai* $\big[d[u], f[u]\big]$ *ir* $\big[d[v], f[v]\big]$ *nesikerta;*

(ii) $\big[d[u], f[u]\big] \subset \big[d[v], f[v]\big]$ *ir u bei v yra viename jungiančio miško medyje, be to, u šiame medyje pasirodo vėliau nei v (u yra v palikuonis);*

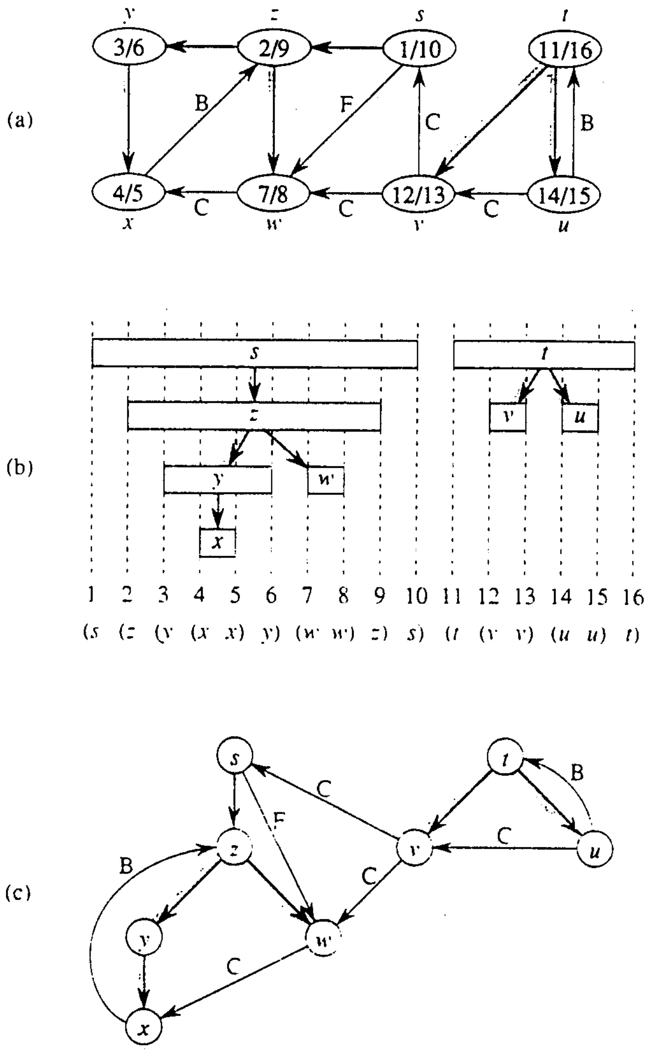
(iii) $\big[d[v], f[v]\big] \subset \big[d[u], f[u]\big]$ *ir u bei v yra viename jungiančio miško medyje, be to, v tokiame medyje pasirodo vėliau nei u (v yra u palikuonis).*

Įrodymas. Galima tarti, kad $d[u] < d[v]$. Jei turėtume, kad ir $d[v] < f[u]$, tai reikštų, jog v buvo atrasta iki u nagrinėjimo pabaigos. Pagal algoritmo strategiją v turėjo būti baigta nagrinėti anksčiau. Vadinasi, $f[v] < f[u]$ ir turime atvejį (iii). Be to, v yra u palikuonis.

Jei $d[v] > f[u]$, intervalai nesikirstų ir turėtume atvejį (i).

Kai $d[u] > d[v]$, pakartojami tie patys samprotavimai. $\diamond$

Paieškos į gylį algoritmo savybės atspindėtos šioje schemoje:



Akcentuokime tokį patogų "palikuonio" kriterijų:

Išvada. *Viršūnė $v \in V$ yra $u \in V$ palikuonis jungiančiajame miške tada ir tik tada, kai*

$$d[u] < d[v] < f[v] < f[u].$$

Kitas būsimų palikuonių atpažinimo kriterijus yra gaunamas iš šios lemos.

2 lema (balto tako l.). *Viršūnė $v \in V$ bus $u \in V$ palikuonis jungiančiajame miške tada ir tik tada, kai momentu $d[u]$ egzistuoja baltų viršūnių $u - v$ takas.*

Įrodymas. Tegu v yra $u \in V$ palikuonis. Abi viršūnės priklausys tam pačiam medžiui, gautam realizavus algoritmą. Bet kuri šio tako viršūnė $w \neq u$ taip pat yra u palikuonis. Vadinasi, laiko momentu $d[u]$ ji bus neatrasta, todėl dar balta.

Tegu dabar yra momentas $d[u]$ ir mes galėtume patekti į v iš u eidami baltosiomis viršūnėmis. Tarkime, kad vykdant *P-Gyl*(G) viršūnė v, deja, nepateko į tą patį medį kaip u. Galim tarti, jog v yra pirmoji tokia viršūnė baltajame take, o prieš tai buvusi balta viršūnė w buvo atrasta ir todėl pateko į jungiančiojo miško medį kartu su u. Vadinasi, $f[w] < f[u]$, kai $w \neq u$. Galėjo būti ir $w = u$. Bet $v \in Adj[w]$, todėl ji turėjo būti atrasta ir apdorota iki momento $f[w]$. Laiko intervalams gauname

$$\big[d[v], f[v]\big] \subset \big[d[w], f[w]\big] \subset \big[d[u], f[u]\big].$$

Suskliaudimo lema sako, kad v yra u palikuonis. $\diamond$

Grafo vidinių savybių tyrimui, pvz., jo cikliškumo nustatymui, reikalinga jo briaunų klasifikacija, išplaukianti iš *P-Gyl*(G) algoritmo realizacijos. Apibrėšime keturias briaunų rūšis.

Briauna uv, patekusi į vieną iš jungiančiojo miško medžių, kai v buvo atrasta tuoj po u, vadinama *medžio* briauna. Kilpa arba briauna uv, kai v buvo protėvis viršūnei u kažkokiame medyje, vadinama *grįžtančiąja. Tiesioginė* briauna uv jungia protėvį u su palikuoniu v. Likusios briaunos vadinamos *skersinėmis.* Jos jungia skirtingus medžius arba vieno medžio viršūnes, kurios nėra viena kitai protėviais arba palikuonimis.

Prisiminę spalvos atributą, galime pastebėti tokius briaunų klasifikavimo kriterijus. Tegu v buvo pastebėta *P-Gyl*(G) metu po u einant briauna uv, tai

(i) jei v balta, tai uv yra medžio briauna;
(ii) jei v pilka, tai uv – grįžtanti;
(iii) jei v juoda, tai uv – tiesioginė arba skersinė briauna.

3 lema. *Neorentuotame grafe kiekviena briauna yra arba medžio briauna, arba grįžtanti.*

Įrodymas. Trivialus. $\diamond$

Algoritmą pritaikykime digrafo savybėms tirti.

Teorema. *Digrafas yra beciklis tada ir tik tada, jei P-Gyl*(G) *neduoda grįžtančių briaunų.*

Įrodymas. Jei yra grįžtanti, tai ciklo egzistavimas akivaizdus.

Tegu digrafas turi ciklą C. Tegu u ir v priklauso jam ir v yra pirmoji ciklo viršūnė, kuri turi būti atrasta einant iš u briauna uv. Momentu $d[v]$ yra baltų viršūnių takas iš v į u prieš ciklo kryptį, kuris lygus $C - uv$. Balto tako lema leidžia tvirtinti, kad u buvo atrasta vėliau nei v, t.y. u yra v palikuonis. Vadinasi, briauna uv yra grįžtančioji. $\diamond$

4. Topologinis rūšiavimas

Nagrinėkime beciklius digrafus. Norime surikiuoti visas viršūnes tiesėje taip, kad visos jas jungiančios briaunos eitų iš kairės į dešinę. *P-Gyl*(G) atlieka šią užduotį.

Teorema. *Jei $f[u]$ yra viršūnių apdorojimo laiko pabaigos atributai, tai viršūnių numeracija su savybe*

$$f[u_1] > \cdots > f[u_n]$$

duoda topologinį rūšiavimą.

Įrodymas. Reikia įsitikinti, kad kiekvienai digrafo briaunai uv turime $f[v] < f[u]$. Jei einant iš u į v viršūnė v yra pilka, tai uv yra grįžtanti briauna. Pagal prieš tai buvusią teoremą, grafas turėtų ciklą. Prieštara sąlygai rodo, kad v yra juoda arba balta. Pirmu atveju, $f[v] < f[u]$ pagal juodo spalvinimo taisyklę. Antruoju atveju v yra palikuonis ir skliaustų lema duoda tą pačią išvadą. $\diamond$

5. Stipriai jungios komponentės

Dirbtinio intelekto uždaviniuose tenka išskirti digrafo $G = (V, E)$ pografius $G' = (V', E')$, ($V' \subset V$ ir $E' \subset E$), kurie turi savybę: bet kokias dvi viršūnes iš V' jungia takai $u \rightsquigarrow v$ bei $v \rightsquigarrow u$. Pastarąjį tako žymėjimą naudojame, jeigu yra einama jo briaunų kryptimis. Ieškoma tokių didžiausios eilės pografių. Pateiksime keletą grafų teorijoje nusistovėjusių sąvokų.

Apibrėžimas. *Pografis $G' = (V', E') < G = (V, E)$ vadinamas indukuotuoju, jei briaunų aibę E' sudaro visos briaunos iš E, kurios jungia V' viršūnių poras.*

Apibrėžimas. *Didžiausios eilės indukuotasis pografis $G' = (V', E')$, tenkinantis sąlygą: kiekvienai porai $u, v \in V'$ turi takus $u \rightsquigarrow v$ bei $v \rightsquigarrow u$, vadinamas stipriai jungia komponente (SJK).*

Pastebėkime, kad yra teisinga tokia lema.

1 lema. *Jei u ir v priklauso stipriai jungios komponentės C viršūnių aibei $V(C)$, tai visos takų $u \rightsquigarrow v$ bei $v \rightsquigarrow u$ viršūnės taip pat jai priklauso.*

Įrodymas. Tegu $u \rightsquigarrow w \rightsquigarrow v$. Pagal SJK apibrėžimą egsistuoja takas $v \rightsquigarrow u$. Vadinasi, $u \rightsquigarrow w$ ir $w \rightsquigarrow u$ per v. Todėl u ir w priklauso C. $\diamond$

Dažnai naudojamas *komponenčių digrafas*, gaunamas sutraukiant SJK į vieną viršūnę. Pastebėkite, kad komponenčių digrafas yra beciklis.

Apibrėžimas. *Digrafo $G = (V, E)$ transpozicija vadinamas grafas $G^T = (V, E^T)$, čia briaunų aibė E^T turi savybę: $uv \in E^T$ tada ir tik tada, jei $vu \in E$.*

Aišku, kad SJK grafe G ir G^T turi tą pačią viršūnių aibę. Kaip rasti SJK-tes? Aišku, kad pakaks rasti jų viršūnių aibių rinkinį. Išnagrinėsime Tarjan'o pasiūlytą algoritmą, kuris remiasi *P-Gyl*(G) procedūra.

SJK(G):

1. Iškviesk *P-Gyl*(G) ir rask $f[u]$ kiekvienai $u \in V$.
2. Rask G^T ir sunumeruok viršūnes $f[u]$ mažėjimo tvarka.
3. Iškviesk *P-Gyl*(G^T) ir rask jungiantįjį mišką.
4. Jungiančiųjų medžių viršūnių aibės yra SJK-čių viršūnių rinkiniai.

Pastebėkime, kad 2 žingsnyje beciklio G^T atveju atliekamas jo topologinis surūšiavimas. Visos procedūros vykdomos per $O(n + m)$ laiko vienetų.

Reikia įrodyti, kad šis algoritmas yra korektiškas.

Teorema. *SJK(G) suranda SJK-čių viršūnių aibes (pačias SJK).*

Įrodymą skaldysim į keletą paprastesnių tvirtinimų.

2 lema. *Vykdant P-Gyl(G), visos vienos SJK-tės viršūnės patenka į vieną jungiančiojo miško medį.*

Įrodymas. Tegu u yra pirma atrasta viršūnė iš SJK-tės. Momentu $d[u]$ egzistuoja balti takai į visas šios komponentės viršūnes. Baltojo tako lema tvirtina, kad jos pateks į tą patį medį. $\diamond$

Labai reta, bet patogi sąvoka pateikiama sekančiame apibrėžime.

Apibrėžimas. *Viršūnės $u \in V$ pirmtaku vadinama $\Phi(u) \in V$, jei egzistuoja $u \rightsquigarrow \Phi(u)$ ir atributas $f[\Phi(u)]$ yra maksimalus.*

Gali būti, kad $\Phi(u) = u$. Pirmtako savybės:

(i) $f[u] \leq f[\Phi(u)]$.

Įrodymas. Akivaizdu. $\diamond$

(ii) Jei $u \rightsquigarrow v$ egzistuoja, tai $f[\Phi(v)] \leq f[\Phi(u)]$.

Įrodymas. Pasiekiamų iš u viršūnių aibė yra nesiauresnė nei iš v. Todėl ir atributo f maksimumas yra nemažesnis. $\diamond$

(iii) $\Phi(\Phi(u)) = \Phi(u)$.

Įrodymas. Kadangi $u \rightsquigarrow \Phi(u)$, tai pagal (ii) $f[\Phi(\Phi(u))] \leq f[\Phi(u)]$. Pagal (i) turime $f[\Phi(u)] \leq f[\Phi(\Phi(u))]$. Iš lygybės $f[\Phi(u)] = f[\Phi(\Phi(u))]$ išplaukia tvirtinimas. $\diamond$

Iš anksto pasakysime, kad SJK-tės viršūnės turi tą patį pirmtaką. Jis bus G^T vieno iš jungiančiojo miško medžių šaknų.

3 lema. *Vykdant P-Gyl(G), bet kokios viršūnės u pirmtakas $\Phi(u)$ yra ir jos prosenelis, t.y. medyje eina prieš u.*

Įrodymas. Jei $\Phi(u) = u$, įrodyta.

Jei ne, nagrinėkim viršūnes momentu $d[u]$. Yra trys atvejai:

a) $\Phi(u)$ yra juoda. Todėl $f[\Phi(u)] < f[u]$. Tai prieštarauja (i) savybei.

b) $\Phi(u)$ yra pilka. Dabar $\Phi(u)$ yra u prosenelis. Tvirtinimas įrodytas.

c) $\Phi(u)$ yra balta. Nagrinėjam taką $u \rightsquigarrow \Phi(u)$. Jei visos jo viršūnės būtų baltos, tai pagal balto tako lemą $\Phi(u)$ yra u palikuonis. Bet tada $f[\Phi(u)] < f[u]$. Tai prieštarautų (i). Vadinasi, take turėtų būti paskutinė nebalta viršūnė, sakykim w. Pastarosios spalva negali būti juoda, nes niekada nėra briaunų iš juodo viršūnės į baltą. Jei w pilka, tai egzistuoja baltas takas nuo jos iki $\Phi(u)$, kuri bus w palikuonis. Ir vėl gauname $f[\Phi(u)] < f[w]$. Tai prieštarautų $f[\Phi(u)]$ maksimalumui. Išnagrinėję visus atvejus baigėme įrodymą. $\diamond$

Išvada. *Viršūnės u ir $\Phi(u)$ priklauso tai pačiai SJK-tei.*

Įrodymas. Pagal apibrėžimą turime taką $u \rightsquigarrow \Phi(u)$. Pagal 3 lemą gavome, kad $\Phi(u)$ yra prosenelis, todėl egzistuoja takas $\Phi(u) \rightsquigarrow u$. $\diamond$

4 lema. *Digrafe $G = (V, E)$, u ir v yra vienoje SJK tada ir tik tada, jei turi tą patį pirmtaką.*

Įrodymas. Tegu $\Phi(u) = \Phi(v)$. Ką tik įrodyta išvada sako, kad u ir $\Phi(u)$ priklauso vienai SJK. Panašiai, ir su v. Tad, abi jos priklauso vienai SJK.

Jei abi viršūnės yra vienoje komponentėje, tai egzistuoja takai $u \rightsquigarrow v \rightsquigarrow u$. Iš čia ir pirmtako apibrėžimo išplaukia lygybė $f[\Phi(u)] = f[\Phi(v)]$. Vadinasi, ir šių viršūnių pirmtakas yra tas pats. $\diamond$

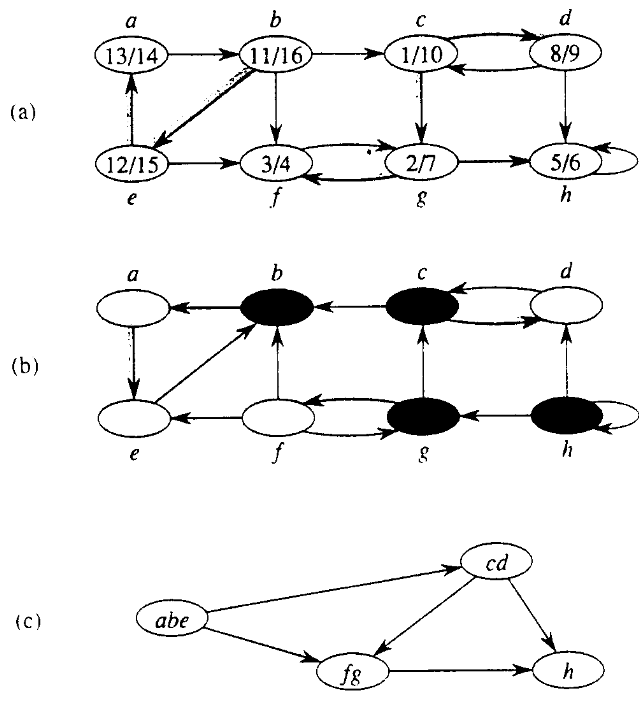
Išvada. *Vykdant* **P-Gyl***(G) SJK-j jos pirmtakas bus pirma atrasta viršūnė ir paskutinė apdorota viršūnė.*

Anksčiau suformuluotos teoremos įrodymas remiasi lemomis.

Teoremos įrodymas. Naudojame indukciją pagal jungiančiojo miško medžių skaičių. Tegu G^T turi vieną medį. Įrodykime, kad tai viena SJK-ė. Prisimename, kad digrafo transpozicija nekeičia SJK-ių. Digrafe G^T viršūnė r su maksimaliu atributu $f[r]$ bus pirmoji atrasta šio digrafo viršūnė ir jungiančiojo miško (dabar iš vieno medžio) šaknis. Iš jos mes pasiekiame kitas šio medžio viršūnes. Vadinasi, digrafe G ši šaknis yra ir pirmtakas, nes jos apdorojimo laikas yra ilgiausias ir ji pasiekiama einant iš kitų medžio viršūnių. Viršūnė r yra joms pirmtakas, tad jos guli vienoj SJK-ėje. Pagal 2 lemą visos SJK, turinčios tą patį primtaką, digrafe G^T patenka į medį su šaknimi r. Tad medis sutampa su SJK-e.

Tegu tvirtinimas yra teisingas digrafams su $s-1$ medžiu. Tegu $r \in V$ yra tokia, kad $f[r] = \max_{v\in V} f[v]$, o
$$C(r) = \{v \in V : \Phi(v) = r\}.$$
Tai viršūnių su bendru pirmtaku aibė. Pagal 4 lemą jos sudaro vienos SJK-ės viršūnių aibę. Pagal aukščiau pateiktus samprotavimus digrafe G^T jos sudarys vieną jungiančiojo miško medį. Išskyrę jį, nagrinėjame digrafą $G - C(r)$, kurio transpozicija turės vienu medžiu mažiau. Pritaikę indukcinę prielaidą, baigiame teoremos įrodymą. $\diamond$

Išsinagrinėkime algoritmo veikimą pagal šią schemą.



Antrasis grafas yra gautas komponenčių grafas.

6. Minimalieji jungiantys medžiai

Nagrinėsime svorinį grafą $G = (V, E)$ su svorio funkcija $w : E \to \mathbf{R}^+$. Tarkime, kad jis yra jungus. Jei $A \subset E$, tai pažymėkime

$$w(A) = \sum_{uv \in A} w(uv)$$

ir vadinkime briaunų aibės A svoriu. Problema:

Rasti jungiantį medį su mažiausiu jo briaunų svoriu.

Tokį medį vadinsime *minimaliuoju jungiančiu medžiu* (MJM). Pastebėkime, jog ne visada godumas padeda: nebūtinai $(n-1)$-a lengviausia briauna sudarys MJM. Algoritmai, kuriuos dabar išnagrinėsime, remiasi bendra strategija, paremta tokiu apibrėžimu:

Apibrėžimas. *Tegu A priklauso MJM. Briauna uv, nepriklausanti A, vadinama saugia dėl A, jeigu $A \cup \{uv\}$ priklauso kažkokiam, gal būt, kitam MJM.*

Minėta strategija yra plėtimas briaunų poaibių panaudojant saugias briaunas. Visiems šiems algoritmams bendra procedūra yra tokia:

B-MJM(G, w):

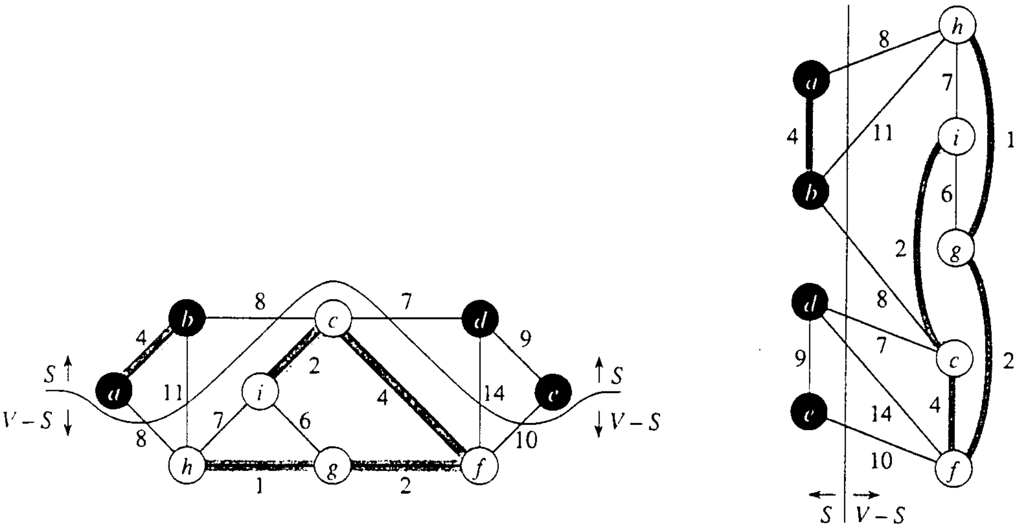
1. $A \leftarrow \emptyset$
2. **while** A nesuformuoja jungiančio medžio
3. **do** rask saugią dėl A briauną uv
4. $A \leftarrow A \cup \{uv\}$
5. RETURN A

Aišku, kad išmestasis A bus MJM, nes "saugumo dėl A" invariantas garantuos jo svorio minimalumą. Žingsnio 3 realizacija ir skirs algoritmus.

Apibrėžimas. *Grafo $G = (V, E)$ pjūvis yra skaidinys $V = S \cup (V - S)$.*

Pjūvį žymėsime $(S, V - S)$. Briaunos iš S į $V - S$ vadinsis *pjūvio briaunomis.* Sakysime, kad pjūvis *nepjauna* $A \subset E$, jeigu jokia briauna iš A nėra pjūvio briauna.

Pjūvius matome šiuose brėžiniuose:

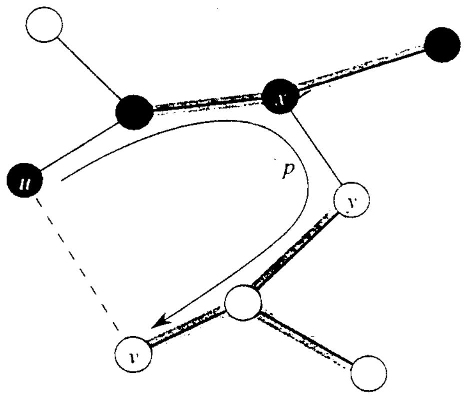


(a) (b)

Apibrėžimas. *Lengva pjūvio briauna vadinsime minimalaus svorio pjūvio briauną.*

Lema. *Tegu A priklauso MJM-iui ir $(S, V - S)$ yra pjūvis, nepjaunantis A. Jei uv yra lengva to pjūvio briauna, tai uv yra saugi dėl A.*

Įrodymas. Tarkime S sudaro juodos viršūnės, o $V - S$ – baltos, kaip parodyta paveiksle. Jame ištisinėmis linijomis nurodytos MJM-io T briaunos. Tegu jam priklauso ir mūsų briaunų aibė A. Jos visų brianų galai yra juodi arba balti. Briauna uv yra lengva ir priklauso pjūviui, todėl nepriklauso aibei A ir T.



Imkime pjūvio briauną xy ir sudarykime

$$T' = T - xy + uv.$$

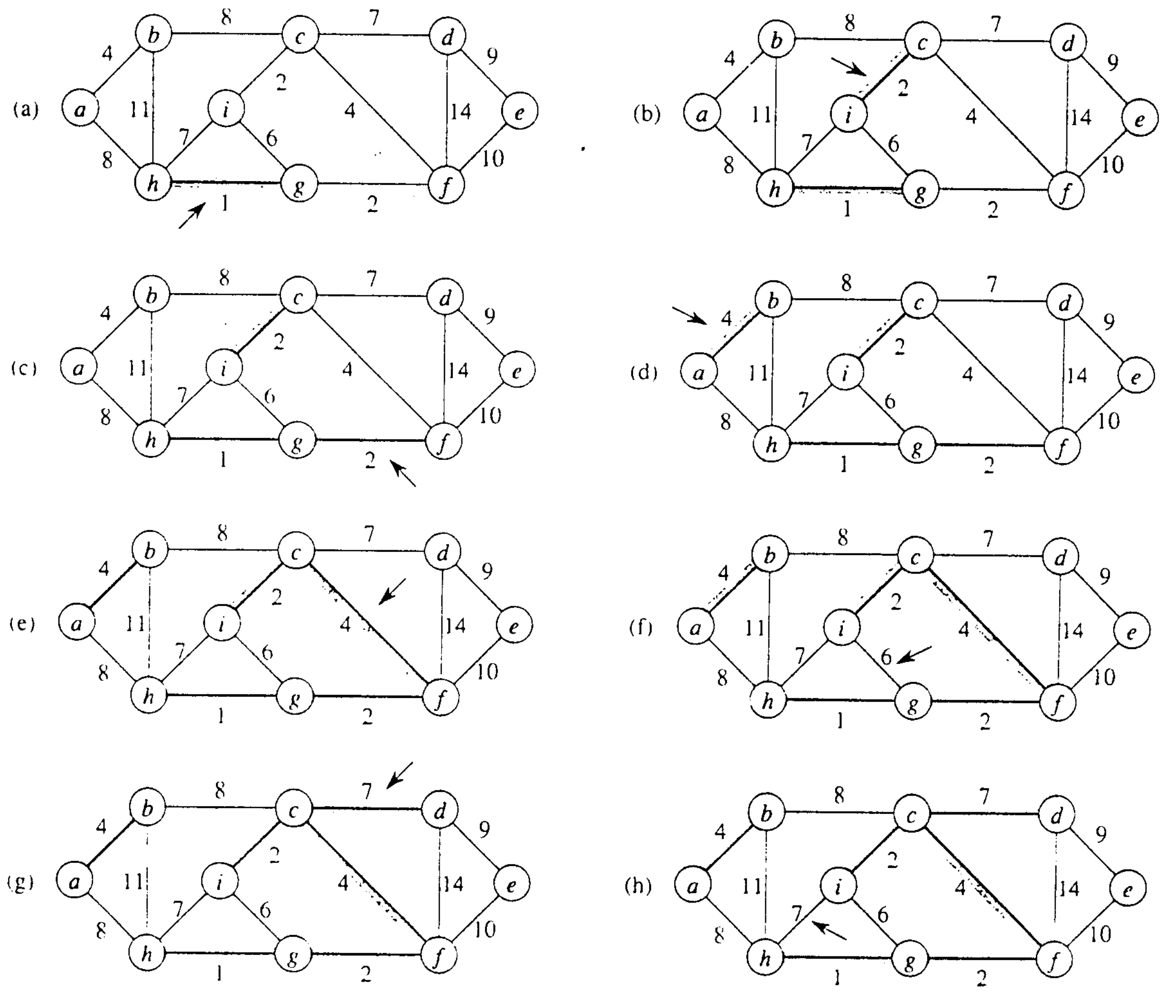
Tai naujas jungiantis medis. Jo svoris

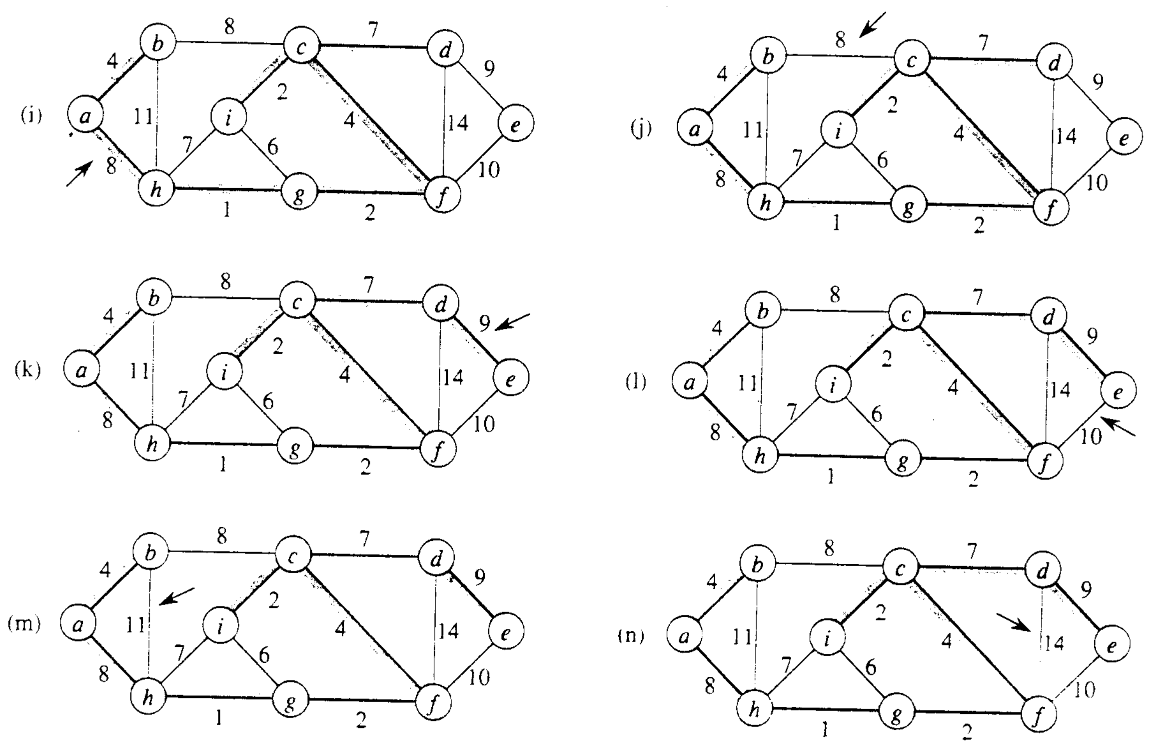
$$w(T') = w(T) - w(xy) + w(uv) \leq w(T),$$

nes uv buvo lengva pjūvio briauna. Todėl ir T' yra MJM-is. Kadangi $A \cup \{uv\} \subset T'$, tai uv yra saugi dėl A. $\diamond$

Lema duoda galimybę pasinaudoti pjūviais ir jų lengvomis briaunomis. Nesunku suvokti, kad MJM-į galima užauginti nuo tuščios aibės A prijungiant tokias briaunas. Pradžioje turėtume mišką iš n medžių po vieną viršūnę, o kaskart prijungiant briauną medžių skaičius mažėtų, naudojami pjūviai turėtų nepjauti medžių, o jų briaunos jungtų du medžius. Lengvomis briaunomis pildydami mišką $G_A = (V, A)$ po $(n-1)$-o žingsnio baigtume MJM-io formavimą.

Kruskalio algoritmo vykdymą matome iš šios schemos:





Paprasčiausias **Kruskal'io algoritmas** generuoja laikinas viršūnių aibes $S(v)$ jas lygina ir apjungia. Todėl jį naudojant reikia žinoti šių procedūrų atlikimo algoritmą. Iš tiesų, $S(v)$ bus aibė viršūnių medžio, kuriam priklauso pati v.

MJM-Kruskal (G, w):

1. $A \leftarrow \emptyset$
2. **for each** $v \in V$
3.   **do** sukurk $S(v)$
4. surūšiuok E briaunas pagal $w(uv)$ didėjimą
5. **for each** $uv \in E$
6.   **do if** $S(u) \neq S(v)$
7.   **then** $A \leftarrow A \cup \{uv\}$
8.   sukurk $S(u) \cup S(v)$
9. RETURN A

Išmesta A ir bus MJM-is, nes mes vykdėme visus lemos reikalavimus. Pradžioj 3 žingsnyje buvo $S(v) = \{v\}$, o 5-7-ame sujungėme dviejų medžių viršūnių aibes. Briauna uv buvo lengva ir saugi dėl A. Žingsnis 6 kontroliuoja, kad nesujungtume dviejų vieno medžio viršūnių. Kadangi pati ilgiausia procedūra vykdoma 4
eilutėje, algoritmo sudėtingumas yra $O(|E| \log |E|)$.

Kruskalio algoritmas proceso metu formuoja jungiantį mišką iš atskirų medžių. Kitas, **Prim'o algoritmas** augina vieną medį A. Pradėjęs nuo bet kokios viršūnės r, tarpiniuose žingsniuose iš viršūnių, nepriklausančių jau suformuotam medžiui A, jis susidaro prioritetinę viršūnių eilę Q. Todėl prireikia rūšiavimo rakto $key[v]$. Jis bus minimalus svoris briaunos uv, jungiančios v su medžiu. Atradus medyje tą u, ji skelbiama v tėvu $\pi[v]$. Formaliai šis medis yra

$$A = \{v\pi[v] : \ v \in V - r - Q\}.$$

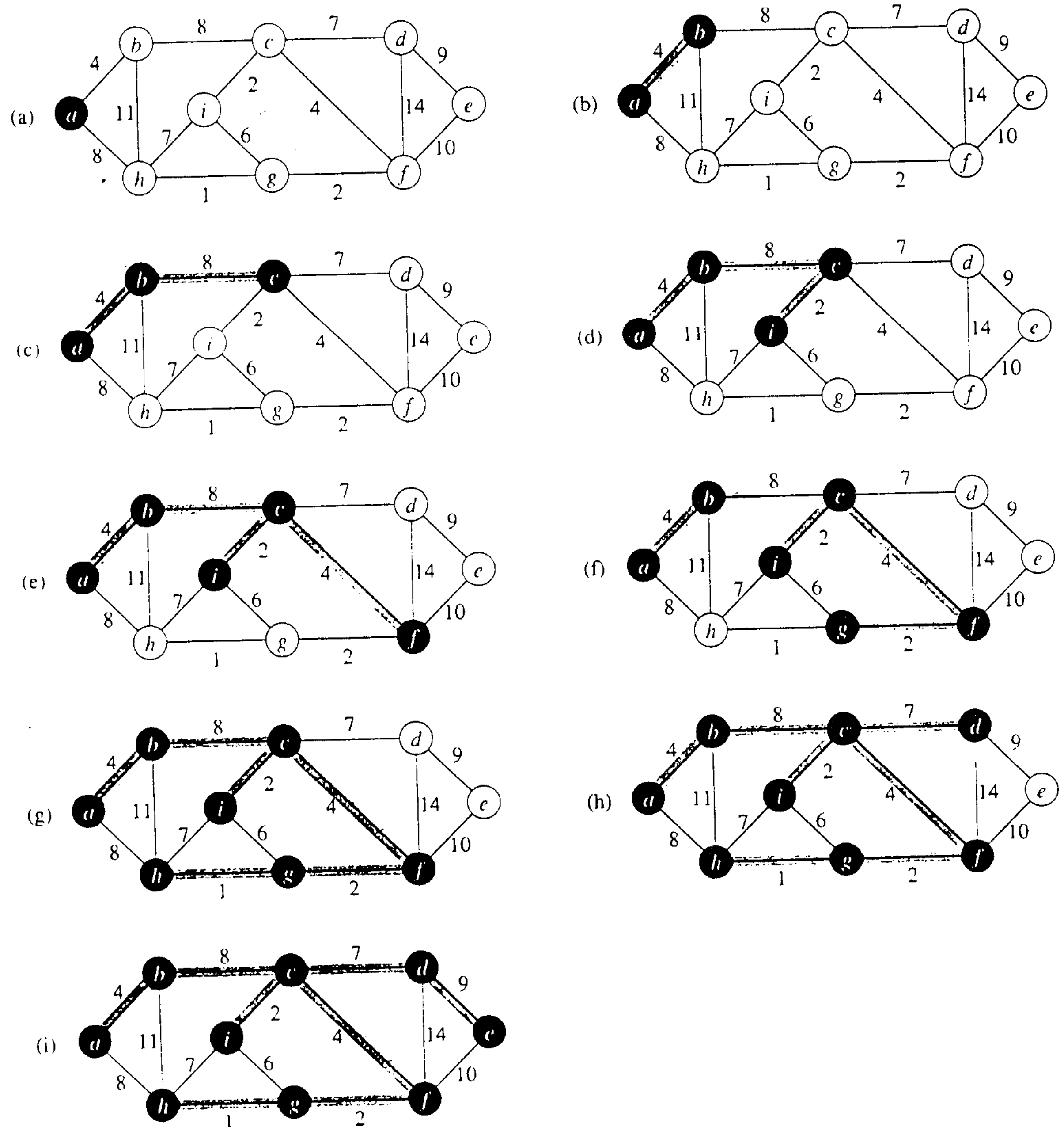
Kai išsemiama eilė Q, algoritmo vykdymas pasibaigia. Medis A yra MJM-is.

MJM-Prim (G, w, r):

1. $Q \leftarrow V$
2. **for each** $u \in Q$
3.   **do** key[u]$\leftarrow \infty$
4. key[r] $\leftarrow 0$
5. $\pi[r] \leftarrow NIL$
6. **while** $Q \neq \emptyset$
7.   **do** $u \leftarrow \min(Q) \leftarrow$ išskirk $\min(Q)$, (be to, ši viršūnė iš Q pašalinama)
8.   **for each** $v \in Adj[u]$
9.   **do if** $v \in Q$ ir $w(uv) < key[v]$
10.   **then** $\pi[v] \leftarrow u$
11. $key[v] \leftarrow w(uv)$
12. RETURN MJM= $\{v\pi[v] : \ v \in V - r\}$

Pastebėkime, kad ir šiame algoritme yra netiesiogiai naudojamas pjūvis $(V - Q, Q)$. Ciklas 8-11 eilutėse randa lengvą pjūvio briauną. Taigi, anksčiau minėta strategija ir čia yra vykdoma. Tai pagal lemą užtikrina algoritmo korektiškumą.

Štai Primo algoritmo veikimo schema:



Algoritmo vykdymo laikas labai priklauso nuo eilės Q apdorojimo. Naudojant krūvas, pasiekiamas $O(m \log n)$ laikas, o Fibonačio krūvas, net – $O(m + n \log n)$ laikas.

7. Trumpiausi atstumai nuo šaltinio

Nagrinėjame svorinį digrafą $G = (V, E)$ su svorio funkcija $w : E \to \mathbf{R}$, kurios reikšmes vadinsime *ilgiais*. Atkreipkime dėmesį, kad ilgiai gali būti ir neigiami. Tai susiję ne tik su atskirais praktikos uždaviniais, bet ir patys algoritmai dažnai tarpiniuose žingsniuose panaudoja neigiamas reikšmes, pvz., einant prieš tako kryptį laikoma, kad įveiktas kelias yra neigiamas. *Kelio ilgiu* vadinsime jo briaunų ilgių sumą. Dabar mus domins trumpiausias tako nuo viršūnės s, vadinamos šaltiniu, iki v ilgis

$$\delta(s,v) = \begin{cases} \min\{w(p):\ p := s \rightsquigarrow v\}, & \text{jei takas egzistuoja,} \\ +\infty \quad \text{kitais atvejais.} \end{cases}$$

Pastebėkime, kad trumpiausias kelio (o ne *tako*, nes jame viršūnės nėra kartojamos) ilgis gali pasiekti net $-\infty$. Taip bus, jeigu iš s pasiektume neigiamo ilgio ciklą, kurioje būtų tikslo viršūnė, ir pirma eitume keliu norimai ilgai. Kai briaunos turi vienetinius ilgius, tako ilgis yra jo briaunų skaičius. Tokią trumpiausio tako problemą sprendėme taikydami paieškos į plotį algoritmą. Jis leido suformuoti trumpiausių takų medį su šaknimi šaltinio viršūnėje. Šįkart pirmiau išdėstysime matematinę trumpiausių atstumų nuo šaltinio paieškos matematinę dalį, o po to paliesime du populiariausius Dijkstros ir Bellmano-Fordo algoritmus.

1 lema. *Jei $G = (V, E)$ yra digrafas, $w : E \to \mathbf{R}$ – svorio funkcija ir $p = v_1 \dots v_k$ – trumpiausias takas $v_1 \rightsquigarrow v_k$, tai kiekvienai porai $1 \le i < j \le k$ takas $v_i \rightsquigarrow v_j$ irgi yra trumpiausias,*

Įrodymas. Turėdami trumpesnį taką iš v_i iš v_j, nepriklausantį p, galėtume ir jį sutrumpinti. ◇

Išvada. *Jei trumpiausias takas $s \rightsquigarrow v$ išsiskaido į taką $s \rightsquigarrow u$ ir briauną $uv \in E$, tai*

$$\delta(s,v) = \delta(s,u) + w(uv).$$

2 lema. *Visada turime*

$$\delta(s,v) \le \delta(s,u) + w(uv).$$

Įrodymas. Trivialu. ◇

Vėliau nagrinėjamuose algoritmuose bus naudojami atributai $d[u]$ ir $\pi[u]$. Pirmasis reikš trumpiausio atstumo nuo s iki v viršutinį įvertį, o antrasis, kaip ir ankstesniuose algoritmuose, žymės tėvą. Jų *inicializacija* bus vykdoma vienodai.

INIT(G,s):

for each $v \in V$
    do $d[v] \leftarrow \infty$
        $\pi[v] \leftarrow NIL$
$d[s] \leftarrow 0$

Taigi, atributai $d[v]$ ir $\pi[v]$ bus kintantys. Svarbiausia procedūra šiuose algoritmuose yra briaunos *relaksacija*, kuri mažins pirmąjį atributą iki $\delta(s,v)$.

RELAX (uv, w) :

If $d[v] > d[u] + w(uv)$
    then $d[v] \leftarrow d[u] + w(uv)$
        $\pi[v] \leftarrow u$

3 lema. *Iškart po briaunos uv relaksacijos turėsime*

$$d[v] \leq d[u] + w(uv).$$

Įrodymas. Akivaizdu. Griežta nelygybė, kai iškvietus RELAX(uv, w) turime $d[v] < d[u] + w(uv)$ ir atributo keisti nereiks. $\diamond$

Algoritmai skirsis relaksacijos procedūros taikymu briaunoms. Todėl pradžioje verta išsiaiškinti, kaip kinta grafas taikant ją.

4 lema. *Po inicializacijos vykdant relaksaciją, visada*

$$d[v] \geq \delta(s, v), \quad v \in V.$$

Be to, jei $d[v]$ pasiekė reikšmę $\delta(s, v)$, toliau vykdant relaksaciją, ji nebekinta.

Įrodymas. Taikome indukciją pagal relaksacijų atlikimo skaičių. Tik po inicializacijos $d[s] = 0 = \delta(s, s)$ ir $d[v] = \infty$ kiekvienai $v \in V \setminus \{s\}$. Tarkime, kad buvo atlikta kažkoks kiekis relaksacijų, ir dabar tas atliekama su $uv \in E$, o v yra pirma viršūnė, kuriai jau

$$d[v] < \delta(s, v).$$

Atlikę šią relaksaciją, pagal 2 lemą gauname

$$d[u] + w(uv) = d[v] < \delta(s, v) \leq \delta(s, u) + w(uv).$$

Tad, $d[u] < \delta(s, u)$. Bet paskutinė relaksacija nepalietė $d[u]$, o pagal indukcijos prielaidą turėjo galioti $d[u] \geq \delta(s, u)$, gavome prieštarą. Pirmasis lemos tvirtinimas įrodytas.

Pastebėję, kad relaksacija atributą d gali tik mažinti ir pagal įrodytą dalį $d[u] \geq \delta(s, u)$, gauname antrąjį teiginį. $\diamond$

Išvada. *Jei nėra tako $s \rightsquigarrow u$, tai inicializacijoje suteikta reikšmė $d[u] = \infty$ išlieka.*

Įrodymas. Kai minimo tako nėra, $\delta(s, u) = \infty$. Pagal 4 lemą $d[u] = \infty$ taip pat. $\diamond$

5 lema. *Tegu $s \rightsquigarrow u \to v$ su $uv \in E$ yra trumpiausias takas. Tarkime, atlikome inicializaciją ir relaksacijų su briaunomis, tarp kurių buvo RELAX(uv, w). Jei iki jos iškvietimo pasitaikė $d[u] = \delta(s, u)$, tai po iškvietimo visada turime $d[v] = \delta(s, v)$.*

Įrodymas. Pastebėkime, kad pagal 4 lemą, atsiradus lygybei $d[u] = \delta(s, u)$, ji išsaugoma ir toliau. Po uv relaksacijos

$$d[v] \leq d[u] + w(uv) = \delta(s, u) + w(uv) = \delta(s, v)$$

pagal 1 lemos išvadą. Iš čia ir 4 lemos išplaukia 5 lemos teiginys. $\diamond$

Dabar aptarkime prosenelių grafo kitimą vykdant relaksacijas. Kaip ir anksčiau jis sudaromas iš viršūnių, jų tėvų ir jas jungiančių briaunų. Tegu

$$V_\pi = \big\{v \in V : \ \pi[v] \neq NIL\big\}, \qquad E_\pi = \big\{\pi[v]v \in E : \ v \in V_\pi \setminus \{s\}\big\}$$

ir $G_\pi = (V_\pi, E_\pi)$ – prosenelių grafas.

6 lema. *Jei svoriniame digrafe $G = (V, E)$ su $w : E \to \mathbf{R}$ nėra neigiamo ilgio ciklų, pasiekiamų iš s, tai bet kuriuo metu po inicializacijos vykdant relaksacijas, prosenelių grafas yra šakninis medis.*

Įrodymas. Indukcija pagal relaksacijų skaičių. Tik po inicializacijos teiginys akivaizdus. Tegu jau atlikome keletą relaksacijų ir gavome $G_\pi = (V_\pi, E_\pi)$. Tegu jame atsirado ciklas $C = v_0 v_1 \dots v_k$ su $v_0 = v_k$. Galim laikyti, kad ciklas susidarė būtent po RELAX$(v_{k-1}v_k, w)$ atlikimo. Priešingu atveju ciklo briaunas tektų pernumeruoti. Pagal G_π apibrėžimą $\pi[v_i] = v_{i-1}$, $1 \le i \le k$. Kadangi kiekviena ciklo viršūnė turėjo tėvą ir baigtinį atributą $d[v_i]$, tai jos yra pasiekiamos iš šaltinio. Relaksacijos priskyrimai

$$d[v_i] \leftarrow d[v_{i-1}] + w(v_{i-1}v_i)$$

buvo įvykdyti prieš RELAX$(v_{k-1}v_k, w)$. Prieš pat šios procedūros pradžią turėjome

$$d[v_i] = d[v_{i-1}] + w(v_{i-1}v_i), \quad i = 1, \dots, k-1,$$

bet

$$d[v_k] > d[v_{k-1}] + w(v_{k-1}v_k).$$

Sumuodami gauname

$$\sum_{i \le k} d[v_i] > \sum_{i \le k} \big(d[v_{i-1}] + w(v_{i-1}v_i)\big) = \sum_{i \le k} d[v_i] + w(C).$$

Todėl $w(C) < 0$. Prieštara rodo, kad prosenelių grafas yra beciklis. Lieka įrodyti, kad jis jungus.

Tegu $v \in V_\pi$ yra pirma viršūnė, nepasiekama iš s. Ji turi tėvą, o $\pi[v]v \in E_\pi$. Tuo atveju ir $d[v] < \infty$. Pagal 4 lemą ir $\delta(s, v) < \infty$. Vadinasi, ji pasiekiama iš šaltinio.

6 lema yra įrodyta. ◇

Ateityje trumpiausių takų šakniniu medžiu vadinsime $G = (V, E)$ pografį $G' = (V', E')$ su V' – pasiekiamų iš s viršūnių aibe, o E' sudarys trumpiausių iš s briaunos.

7 lema. *Jei svoriniame digrafe $G = (V, E)$ su $w : E \to \mathbf{R}$ nėra neigiamo ilgio ciklų, pasiekiamų iš s, tai bet kuriuo metu po inicializacijos vykdant relaksacijas, kurios davė $d[v] = \delta(s, v)$ kiekvienam $v \in V$, prosenelių grafas yra trumpiusių takų šakninis medis.*

Įrodymas. Pagal 6 lemą jis yra šakninis medis, dabar netgi jungiantysis. Ar jame esantys takai $s \rightsquigarrow v$ yra trumpiausi?

Tegu $s \rightsquigarrow v$ yra $p = s = v_0, v_1, \dots, v_k = v$. Turime

$$d[v_i] = \delta(s, v_i), \quad i = 0, 1, \dots, k;$$

ir

$$d[v_i] = d[v_{i-1}] + w(v_{i-1}v_i), \quad i = 0, 1, \dots, k.$$

Tad,

$$w(v_{i-1}v_i) = \delta(s, v_i) - \delta(s, v_{i-1}), \quad i = 1, \ldots, k.$$

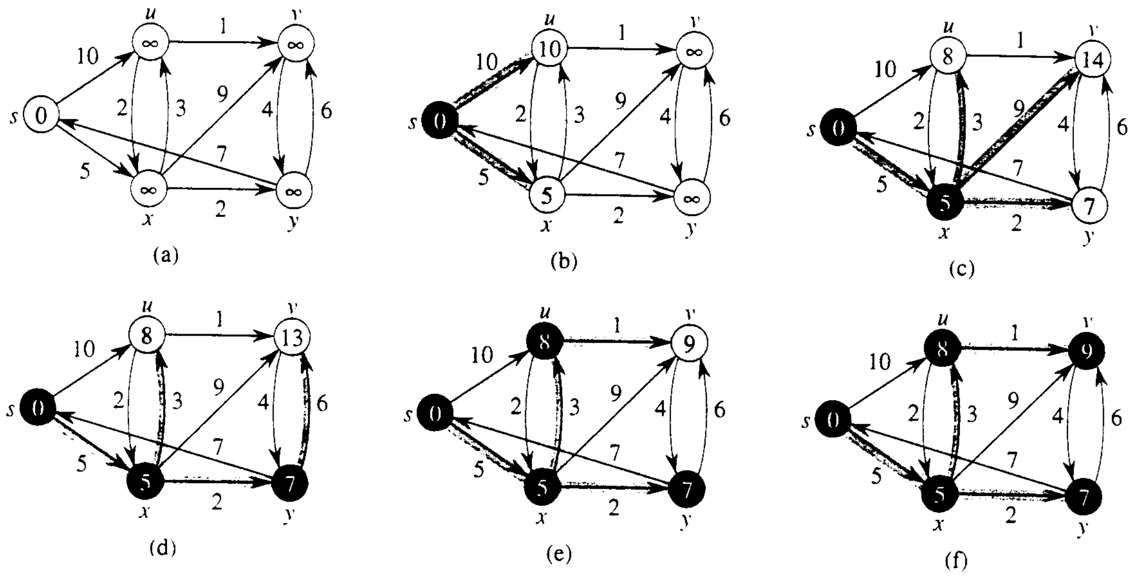
Susumavę gauname $w(p) = \delta(s, v_k)$. Tai rodo, kad šis takas yra trumpiausias. $\diamond$

Dabar pateiksime Dijkstra algoritmą, pasiūlytą 1959 metais. Jo vykdymo eigoje yra formuojama viršūnių aibės poaibis S ir iš likusių viršūnių sudarinėjama prioritetinė eilė Q pagal $d[u]$ didėjimo tvarką.

DIJKSTRA (G, w, s) :

```
INIT(G,s)
S ← ∅
Q ← V
while Q ≠ ∅
      do u ← "išskirk min(Q)"        (ir išmeta iš Q)
            S ← S ∪ {u}
            for each v ∈ Adj[u]
                  do RELAX(uv,w)
END
```

Jo vykdymas pavaizduotas šiame brėžinyje:

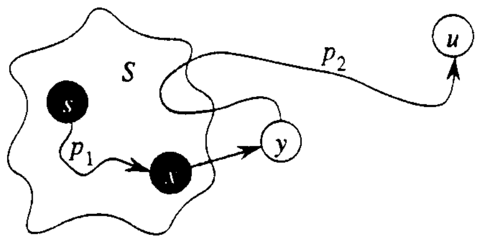


Turėdami visas reikalingas lemas galime įrodyti šio algoritmo korektiškumą.

Teorema. *Tegu $G = (V, E)$ yra digrafas su svorio funkcija $w : E \to \mathbf{R}^+$ ir šaltiniu s. Realizavus Dijkstra algoritmą, kiekvienai $u \in V$ gauname $d[u] = \delta(s, u)$. Sudarytas prosenelių grafas yra trumpiausių takų medis.*

Įrodymas. Iš 4 lemos žinome, kad $d[u] \geq \delta(s, u)$. Įsitikinsime, kad tuo metu, kai u išmetama iš Q ir įrašoma į S, ji pasiekia $\delta(s, u)$.

Tegu u yra pirmoji viršūnė, kuriai taip nėra. Tuo metu, kai ji įrašoma į S, $\delta(s,u) < d[u]$. Vadinasi, u yra pasiekiama iš s. Aišku, $s \neq u$, todėl net prieš įstatant u, aibė S nebuvo tuščia. Tuo momentu egzistuojantis trumpiausias takas $s \rightsquigarrow u$ eina iš aibės S ir patenka į $V - S$. Galime įsivaizduoti situaciją brėžinyje:



Tas takas yra toks: $s \rightsquigarrow x \rightarrow y \rightsquigarrow u$ su $x \in S$, $y \in V - S$ bei $xy \in E$. Gal būt, eidami iš y vėl buvome grįžę į S ar pasitaikė, kad $y = u$ arba $s = x$. Pagal 1 lemą takas $s \rightsquigarrow y$ irgi trumpiausias. Kadangi x yra jau aibėje S, tai $d[x] = \delta(s,x)$. Briaunos xy relaksacija buvo atlikta, tad jei $y = u$, iš 5 lemos gautume $d[u] = \delta(s,u)$. Prieštara rodo, kad $y \neq u$. Pritaikome 5 lemą takui $s \rightsquigarrow y$, einančiam per briauną xy. Gauname

$$d[y] = \delta(s,y) < \delta(s,u) < d[u].$$

Vadinasi, renkant minimumą 5 eilutėje iš Q pirmiau turėjo būti y, o ne u. Prieštara įrodo pirmą teoremos tvirtinimą.

Antrasis teiginys išplaukia iš 6 lemos. $\diamond$

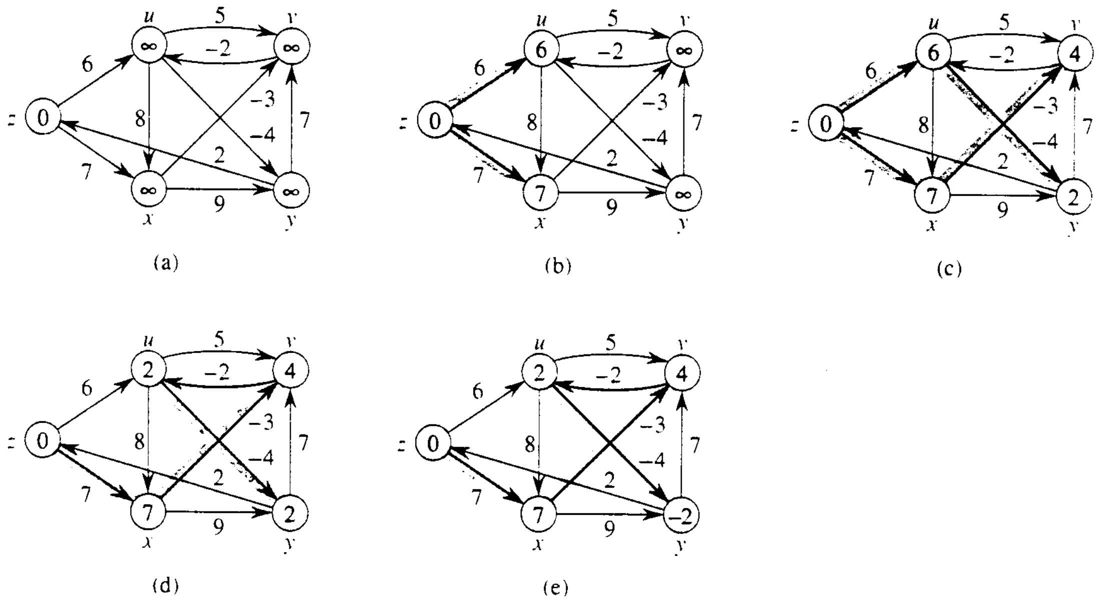
Tirdami algoritmo vykdymo trukmę, pastebime, kad $Q = V - S$ tiesinis masyvas, jame minimumo išrinkimas lengvai realizuojamas per $O(|V|)$ žingsnių. Taikant binariąsias ar Fibonačio krūvas, pakaktų netgi $O(\log |V|)$ žingsnių. Tai taikoma $|V|$ kartų. Kaip visada, gretimumo sąrašo peržiūrėjimas užtrunka $O(|E|)$ laiko vienetų. Tad, gera programa būtų vykdoma $O\big((|V| + |E|) \log |V|\big)$ laiko vienetų.

Bellmano-Fordo algoritmas taikomas ir digrafams su svorio funkcija $w : E \to \mathbf{R}$. Kai egzistuoja neigiamo ilgio ciklai, jame naudojamas kintamasis su reikšmėmis "True" ir "False" nurodo antrąjį atvejį. Kai jis lygus "True", algoritmas duoda trumpiausių takų medį.

B-F(G, w, s):

1. INIT(G, s)
2. **for** $i \leftarrow 1$ **to** $|V| - 1$
3.   **do for each** edge $uv \in E$
4.   **do** RELAX(uv, w)
5. **for each** edge $uv \in E$
6.   **do if** $d[v] > d[u] + w(uv)$
7.   **then return** "False"
8. **return** "True"
9. END

Štai algoritmo vykdymo schema:



Pastebėkime, kad 2 eilutė verčia kartoti relaksacijas iš naujo visoms briaunoms. Vėliau įsitikinsime, kad $|V|-1$ karto pakanka. Bet čia sugaištama $O(|V||E|)$ laiko. Išnagrinėkime algoritmo korektiškumą.

8 lema. *Jei digrafas $G = (V, E)$ su svorio funkcija $w : E \to \mathbf{R}$ neturi neigiamo ilgio ciklų, pasiekiamų iš šaltinio viršūnės s, tai pasibaigus Bellmano-Fordo algoritmo vykdymui, kiekvienai iš s pasiekiamai viršūnei v gauname $d[v] = \delta(s, v)$.*

Įrodymas. Tarkime v yra pasiekiama ir $p = v_0 v_1 \dots v_k$ yra trumpiausias takas iš $s = v_0$ iki $v = v_k$. Aišku, kad $\delta(s, v) = k \leq |V| - 1$. Naudodami indukciją parodysime, kad po i

algoritmo 2 eilutėje nurodytų iteracijų mes gausime

$$d[v_i] = \delta(s, v_i).$$

Kai $i = 0$, tas išplaukia iš inicializacijos. Be to, 4 lema sako, kad vėlesnės relaksacijos pasiektos lygybės $\delta(s, v_j) = d[v_j]$ nebekeičia.

Tegu indukcinė prielaida, kad

$$d[v_{i-1}] = \delta(s, v_{i-1}).$$

buvo pasiekta po $(i-1)$-os iteracijos, yra teisinga. Briauna $v_{i-1}v_i$ yra relaksuojama i žingsnyje. Pasinaudoję 5 lema, gauname, kad jame pasiekiamas apatinis $d[v_i]$ rėžis $\delta(s, v_i)$. Jis pagal 4 lemą vėliau nebekinta. ◇

Pastebėkime, kad nepasiekiamų viršūnių atributas $d[v] = \infty$ išlieka, nes iškvietus *RELAX* procedūrą yra tikrinama nelygybė briaunos galų atributams. O nepasiekiamai viršūnei tokios briaunos, tenkinančios šią sąlygą, neturėsime.

2 teorema (B-F algoritmo korektiškumas). *Jei digrafas $G = (V, E)$ su svorio funkcija $w : E \to \mathbf{R}$ neturi neigiamo ilgio ciklų, pasiekiamų iš šaltinio viršūnės s, tai pasibaigus Bellmano-Fordo algoritmo vykdymui, kiekvienai iš s pasiekiamai viršūnei v gauname $d[v] = \delta(s, v)$, prosenelių pografis yra trumpiausių takų medis ir užrašoma "True". Jei digrafas G turi neigiamo ilgio ciklų, tai pasirodo "False".*

Įrodymas. Pirmuoju atveju, jei digrafas $G = (V, E)$ su svorio funkcija $w : E \to \mathbf{R}$ neturi neigiamo ilgio ciklų, pasiekiamų iš šaltinio viršūnės s, tai 8 lema įrodo lygybę $d[v] = \delta(s, v)$. Pasiekiamoms viršūnėms šis dydis yra baigtinis, o nepasiekiamoms – begalinis. Pagal 6 lemą suformuotas prosenelių pografis yra trumpiausių takų į pirmojo tipo viršūnes medis.

Įvykdžius algoritmą, jei $uv \in E$,

$$d[v] = \delta(s, v) \le \delta(s, u) + w(uv) = d[u] + w(uv).$$

Todėl algoritmo 8-oje eilutėje nurodyta sąlyga yra nepatenkinta ir turi būti išmetamas užrašas "True".

Tegul turime neigiamo ilgio ciklą $C = v_0v_1 \ldots v_k$, $v_0 = v_k$ su $w(C) < 0$, kuris yra pasiekiamas iš s. Jei vis tik pasirodo reikšmė "True", turėjo būti

$$d[v_i] \le d[v_{i-1}] + w(v_{i-1}v_i)$$

kiekvienam $i = 1, \ldots, k$. Susumavę šias nelygybes, gautume prieštarą: $w(C) \ge 0$.

Teorema įrodyta. ◇

Pastebėkime, kad becikliams digrafams Dijkstra ir Bellmano-Fordo algoritmus galima pagerinti pasinaudojus anksčiau nagrinėtu topologiniu rūšiavimu. Kaip nurodyta 4 skyrelyje, tokių digrafų viršūnes galima išrikiuoti taip, kad visos digrafo briaunos eitų iš kairės į dešinę. Panaudojus paieškos gilyn algoritmą, suradus viršūnių apdorojimo pabaigos laiko atributus $f[u]$, pakanka viršūnes išrikiuoti šio atributo mažėjimo atžvilgiu. Topologinis rūšiavimas užtrunka $O(|V|+|E|)$ laiko.

Trumpiausių takų nuo šaltinio iki visų viršūnių pagerintas algoritmas simboliškai vykdomas tokiais žingsniais:

B-F-TrT (G, w, s):

1. Atlikti topologinį rūšiavimą
2. INIT (G, s)
3. **for each** u iš topologiškai surūšiuoto sąrašo (gal būt, net ne s)
4. **do for each** $v \in Adj[u]$
5. **do** $RELAX(uv, w)$
6. END

INSERT 537 psl.

Lyginant su Dijkstra algoritmu, nebereikia išskirti $\min(Q)$, kas užtrunka $O(\log|V|)$ laiko ir kartojama $|V|$ kartų. Dabar bendras relaksacijų skaičius yra trumpesnis. Šio algoritmo vykdymas užtrunka tik $O(|V|+|E|)$ laiko.

3 teorema. *Realizavus šį algoritmą, atributai $d[u]=\delta(s,u)$ kiekvienai $u \in V$. Sudarytas prosenelių pografis yra trumpiausių takų medis.*

Įrodymas. Pastebėkime, kad nepasiekiamų iš s viršūnių gali būti kairėje nuo s. Bet joms po inicializacijos suteiktas $d[u]=\infty$ nepakis, kaip ir kitoms nepasiekiamoms viršūnėms.

Pasiekiamoms viršūnėms, suradę atstumą iki pirmosios viršūnės trumpiausiame take $p = v_0v_1\dots v_k$, $v_0 = s$, einame toliau į dešinę. Paprastas indukcijos panaudojimas, paremtas lygybe

$$\delta(s, v_i) = \delta(s, v_{i-1}) + w(v_{i-1}v_i),$$

įrodo teoremos teiginį. Trivialūs samprotavimai pagrindžia ir trumpiausių takų medžio konstrukciją. $\diamond$

Keletas žodžių pasakytina apie šiame skyrelyje nagrinėjamų algoritmų taikymus. Jie dažnai naudojami testuojant programas. Jose tam tikri momentai (etapai) laikomi digrafo viršūnėmis, o tarpinės procedūros pereinant nuo vieno momento prie kito, kai yra atliekama tam tikra procedūra – briaunomis. Procedūros vykdymo laikas laikomas briaunos svoriu. Tokie digrafai neturi ciklų, todėl galimas visų trijų algoritmų panaudojimas su tikslu rasti trumpiausius atskirų užduočių įvykdymo laikotarpius.

Apsistosime ties labai specialiu tiesinio programavimo uždavinio epizodu. Tirsime nelygybių

$$AX \le B \tag{1}$$

atskirųjų sprendinių egzistavimą. Čia $A = (a_{ij})$, $1 \le i \le m$, $1 \le j \le n$, yra reali matrica, B – matrica-stulpelis $n \times 1$ ir X – nežinomųjų x_j vektorių stulpelis. Kai A kiekvienoje eilutėje lygiai du elementai yra nenuliniai, o ir tie yra 1 ir -1, turime *skirtuminių apribojimų* matricą.

Tada (1) yra tiesiog nelygybių sistema. Pavyzdžiui,

$$x_1 - x_3 \le 0, \qquad x_1 - x_4 \le -1,$$

$$x_2 - x_5 \le 5, \qquad x_2 - x_6 \le 2,$$

$$x_3 - x_2 \le 4, \qquad x_3 - x_5 \le -1,$$

$$x_6 - x_3 \le 4, \qquad x_6 - x_5 \le 3.$$

Kaip nustatyti, ar ši ir panašios skirtuminių apribojimų sistemos turi sprendinių? Pastebėkime, kad suradus vieną sprendinį $X = X_0$, iš karto randame jų be galo daug: $X_0 + D$, čia D – vektorius su vienodomis koordinatėmis, irgi yra sprendinys.

Sudaromas apribojimų digrafas $G_A = (V, E)$. Prie n viršūnių, atitinkančių kiekvieną iš nežinomųjų, yra prijungiama papildoma viršūnė s. Tad, $V = \{s, v_1 \dots, v_n\}$. Jei apibojimų sistemoje yra nelygybė $x_j - x_i \le b_k$, tai digrafe apibrėžiama briauna $x_i x_j$ su svoriu b_k. Be to, išvedamos visos briaunos sv_j, $1 \le j \le n$, ir laikoma, kad $w(sv_j) = 0$ kiekvienam $1 \le j \le n$.

4 teorema. *Jei digrafas G_A neturi neigiamo ilgio ciklo, tai skirtuminių apribojimų sistema (1) turi sprendinį*

$$X_0 = (\delta(s, v_1), \dots, \delta(s, v_n))^t,$$

čia t reiškia vektoriaus transponavimą. Jei digrafas G_A turi neigiamo ilgio ciklą, tai skirtuminių apribojimų sistema (1) neturi sprendinio.

Įrodymas. Pagal 2 lemą pirmuoju teoremos atveju

$$\delta(s, v_j) \le \delta(s, v_i) + w(v_i v_j).$$

Todėl paėmę $x_j^0 = \delta(s, v_j)$ matome, jog

$$x_j^0 - x_i^0 \le w(v_i v_j).$$

Tai ir buvo atitinkamas skirtuminis apribojimas.

Tegu egzistuoja neigiamo ilgio ciklas $C = v_1 \dots v_k$ su $v_1 = v_k$ ir $k \ge 2$. Numeracija neturės įtakos samprotavimams, bet cikle nebus šaltinio viršūnės, nes į ją jokia briauna negrįžta. Ciklo briaunos atitiko skirtuminius apribojimus, todėl

$$x_2 - x_1 \le w(v_1 v_2) \quad , \dots , \quad x_k - x_{k-1} \le w(v_{k-1} v_k), \quad x_1 - x_k \le w(v_k v_1).$$

Sudėję visas nelygybes gauname

$$w(v_1 v_2) + \dots + w(v_{k-1} v_k) + w(v_k v_1) \ge 0.$$

Prieštara įrodo antrąjį teoremos teiginį. $\diamond$

8. Trumpiausi takai tarp visų viršūnių

Vėl nagrinėjame svorinį digrafą $G = (V, E)$ su svorio funkcija $w : E \to \mathbf{R}$. Trumpinant užrašus imsime $V = \{1, \ldots, n\}$. Dabar ieškosime visų atstumų tarp viršūnių porų. Aišku, galime apsinaudoti 7 skyrelio algoritmais laikant kiekvieną viršūnę šaltiniu. Tai pakankamai ilga procedūra. Kai svorio funkcija neneigiama, pritaikę Dijkstra algoritmą $|V|$ kartų, sugaištume $O(|V|^3 + |V||E|) = O(|V|^3)$ laiko, kai prioritetinės eilės minimumas ieškomas netobulai. Naudodami krūvas galėtume pasiekti $O(|V|^2 \log |V| + |V||E|)$ vykdymo laiką. Lėtesnis Bellmano-Fordo algoritmas irgi panaudotinas. Mūsų tikslas – išnagrinėti šiek tiek geresnius šia prasme algoritmus, išvengiant kartotinio ankstesnių procedūrų taikymo. Reikia pasakyti, kad tik retiems grafams greičio pagerinimas yra pasiekiamas.

Dabar patogiau manipuliuoti digrafo briaunų svorių matrica $W = (w_{ij})$, $1 \leq i, j \leq n$ apibrėžiant, kad $w_{jj} = 0$ (G – bekilpis grafas!), $w_{ij} = w(ij)$, kai $i \neq j$ ir $ij \in E$. Jei tokios briaunos nėra, laikoma, kad $w_{ij} = \infty$. Ši matrica yra pirmoji iteracija kintamajai trumpiausių atstumų tarp viršūnių porų matricai $D = (d_{ij})$. Pasibaigus algoritmo vykdymo laikui turėsime $d_{ij} = \delta(i, j)$. Panašiai, formuojama protėvių matrica $\Pi = (\pi_{ij})$, $1 \leq i, j \leq n$, su $\pi_{jj} = NIL$, kai $i = j$, arba, kai j nepasiekiama iš i. Kitais atvejais π_{ij} žymi j viršūnės kažkokį proseneli trumpiausiame $(i - j)$ take. Dabar tenka formuoti n prosenelių pografių. Kiekvienam $1 \leq i \leq n$ pažymėkime

$$G_{\pi i} = (V_{\pi i}, E_{\pi i}),$$

$$V_{\pi i} = \{j \in V : \ \pi_{ij} \neq NIL\} \cup \{i\}$$

ir

$$E_{\pi i} = \{\pi_{ij} j : \ j \in V_{\pi i} \text{ ir } \pi_{ij} \neq NIL\}.$$

Kai nėra neigiamo ilgio ciklų, išnaudojama trumpiausio tako struktūra

$$i \rightsquigarrow k \to j.$$

Jei $i \rightsquigarrow k$ turėjo $(m-1)$-ą briauną, tai pagal anksčiau turėtas lemas $i \rightsquigarrow j$ turės m, be to,

$$\delta(i, j) = \delta(i, k) + w_{ij}.$$

Tai leidžia daryti iteracijas.

Tegu d_{ij}^m yra mažiausias svoris $i \rightsquigarrow j$ tako, kuriame yra ne daugiau kaip m briaunų. Pažymėkime

$$d_{ij}^0 = \begin{cases} 0, \text{ jei } i = j, \\ \infty, \text{ jei } i \neq j. \end{cases}$$

Tarkime, kad jau apskaičiavome d_{ij}^{m-1}, tada

$$\begin{aligned} d_{ij}^m &= \min\big\{d_{ij}^{m-1}, \min_{1 \leq k \leq n} (d_{ik}^{m-1} + w_{kj})\big\} \\ &= \min_{1 \leq k \leq n} (d_{ik}^{m-1} + w_{kj})\big\}. \end{aligned}$$

Čia pasinaudojome lygybe $w_{jj} = 0$. Kadangi take yra ne daugiau negu $(n-1)$-a briauna, gautume

$$d_{ij}^{n-1} = d_{ij}^{n} = \cdots = \delta(i,j).$$

Taigi, paprastame algoritme pakaktų skaičiuoti iteracijų seką

$$W = D^{(1)} \to D^{(2)} \to \cdots \to D^{(n-1)}.$$

Pagrindinė sudėtinė dalis būtų procedūra, kurioje iš dviejų jau žinomų matricų D, W yra apskaičiuojama trečia D'. Štai ji:

Tr-T (D, W):

$n \leftarrow$ eilutės[D] (įveda matricos D eilučių skaičių)
Tegu $D' = (d'_{ij})$ – $n \times n$ matrica
for $i \leftarrow 1$ **to** n
    do for $j \leftarrow 1$ **to** n
        do $d'_{ij} \leftarrow \infty$
            for $k \leftarrow 1$ **to** n
                do $d'_{ij} \leftarrow \min\{d'_{ij}, d_{ik} + w_{kj}\}$
return D'

Procedūra trunka $O(n^3)$ žingsnių. Palyginkime ją su panašiu matricų $A = (a_{ij})$ ir $B = (b_{ij})$ daugybos algoritmu.

M-D (A, B):

$n \leftarrow$ eilutės[A] (įveda matricos A eilučių skaičių)
Tegu $C = (c_{ij})$ – $n \times n$ matrica
for $i \leftarrow 1$ **to** n
    do for $j \leftarrow 1$ **to** n
        do $c'_{ij} \leftarrow 0$
            for $k \leftarrow 1$ **to** n
                do $c_{ij} \leftarrow \{c_{ij} + a_{ik} \cdot b_{kj}\}$
return C

Nesunku įsitikinti, kad pastarasis algoritmas suskaičiuoja matricų A ir B sandaugą. Abiejų procedūrų panašumas akivaizdus. Tiesiog pakanka sumą pirmoje procedūroje pakeisti sandauga, o "min" – suma. Keletą kartų taikant *M-D*(A, A), suskaičiuojame laipsnius. Jei reikia rasti A^n su dideliu n, tai nebūtina daryti n veiksmų. Galime kelti rastą A^2 kvadratu, o vėliau ir patį rezultatą pakelti kvadratu, ir t.t. Pasielkime panašiai ir su *Tr-T*(D, W). Pažymėkime gautus rezultatus atitinkamai:

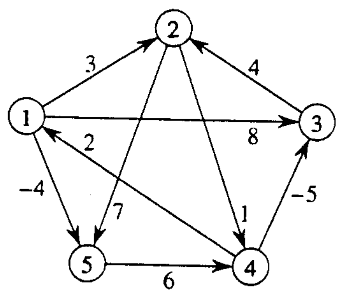
$$\mathbf{Tr-T}(D_0, W) = D^{(1)}, \quad \mathbf{Tr-T}(D^{(1)}, W) = D^{(2)}, \quad \mathbf{Tr-T}(D^{(k)}, W) = D^{(k+1)}, \ldots$$

Šių matricų skaičiavimo procedūra būtų tokia:

Lėta Tr-T(W):

1. $n \leftarrow$ eilutės[W]
2. $D^{(1)} \leftarrow W$
3. **for** $m \leftarrow 2$ **to** $n-1$
4. **do** $D^{(m)} \leftarrow \mathbf{Tr} - \mathbf{T}(D^{(m-1)}, W)$
5. **return** $D^{(n-1)}$

Dabar po $O(n^4)$ žingsnių gavom trumpiausių takų matricą.



$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \qquad D^{(4)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

Pasinaudoję minėta matricų laipsnio skaičiavimo idėja procedūrą sutrumpiname. Radę $D^{(2)}$ toliau tesiame

$$\mathbf{Tr} - \mathbf{T}(D^{(2)}, D^{(2)}) = D^{(4)}, \quad \mathbf{Tr} - \mathbf{T}(D^{(4)}, D^{(4)}) = D^{(2^3)},$$
$$\mathbf{Tr} - \mathbf{T}(D^{(2^k)}, D^{(2^k)}) = D^{(2^{k+1})}, \ldots$$

Kai $2^{k+1} > n-1$, vykdymą galima nutrukti, nes trumpiausių atstumų matrica jau gauta ir nebekis. Vietoje n kartų anksčiau taikytų procedūrų pakaks $O(\log n)$ žingsnių. Todėl greitesnis algoritmas būtų toks:

Greita Tr-T(W):
1. $n \leftarrow$ eilutės[W]
2. $D^{(1)} \leftarrow W$
3. $m \leftarrow 1$
4. **while** $n - 1 > m$
5. **do** $D^{(2m)} \leftarrow \mathbf{Tr} - \mathbf{T}(D^{(m)}, D^{(m)})$
6. $m \leftarrow 2m$
5. **return** $D^{(m)}$

Taigi, jau dabar galėtume rasti trumpiausių takų matricą per $O(n^3 \log n)$ žingsnių. Egzistuoja dar greitesnių algoritmų.

Floyd'o ir Warshall'o algoritmui pakaks $O(n^3)$ žingsnių. Jis išnaudoja kitokią nei iki šiol buvo naudota trumpiausio tako struktūrą. Tako $p = iv_2 \ldots v_{k-1}j$ viršūnes $v_2 \ldots v_{k-1}$ vadinkime *tarpinėmis.* Nagrinėjamasis algoritmas klasifikuoja takus pagal tai, kokios yra tarpinės viršūnės. Ištyręs takus, kurių tarpinės viršūnės priklauso aibei $\{1, \ldots, k-1\}$, imasi takų praplėsdamas šią aibę iki $\{1, \ldots, k-1, k\}$. Pažymėkime trumpiausius takus tarp kažkokių viršūnių i ir j su minėtomis tarpinių viršūnių savybėmis raidėmis p_{k-1} ir p_k ir raskime jų rekurenčiuosius sąryšius. Jei k nebuvo tarpine viršūne take p_k, tai tiesiog $p_k = p_{k-1}$. Priešingu atveju

$$p_k : \quad i \leadsto k \leadsto j,$$

čia esantys daliniai takai jau priklauso takams pažymėtiems raide p_{k-1}. Tuo pasinaudokime.

Tegu d_{ij}^k yra ilgis tako $i \leadsto j$ su tarpinėmis viršūnėmis iš $\{1, \ldots, k\}$. Jei $k = 0$, tai jokių tarpinių viršūnių nėra, todėl arba $i \leadsto j = i \to j$ arba $\delta(i,j) = \infty$. Kitaip sakant,

$$d_{ij}^0 = w_{ij}.$$

Apibrėžiant rekursyviai gauname

$$d_{ij}^k = \begin{cases} w_{ij}, & \text{jei } k = 0, \\ \min\left\{d_{ij}^{k-1},\, d_{ik}^{k-1} + d_{kj}^{k-1}\right\}, & \text{jei } k \geq 1. \end{cases}$$

Kadangi visos tarpinės viršūnės priklauso aibei $V = \{1, \ldots, n\}$, gausime

$$D^{(n)} = (d_{ij}^n) = (\delta(i,j)).$$

Pati Floydo-Warshallo algoritmo procedūra yra tokia:

Fl-W (W):
1. $n \leftarrow$ eilutės[W]
2. $D^{(0)} \leftarrow W$
3. **for** $k \leftarrow 1$ **to** n
4. **do for** $i \leftarrow 1$ **to** n
5. **do for** $j \leftarrow 1$ **to** n
6. $d_{ij}^k \leftarrow \min\left\{d_{ij}^{k-1},\, d_{ik}^{k-1} + d_{kj}^{k-1}\right\}$
5. **return** $D^{(n)}$

Pakanka $O(n^3)$ žingsnių.

Pavyzdyje gaunama tokia matricų seka:

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(0)} = \begin{pmatrix} \mathrm{NIL} & 1 & 1 & \mathrm{NIL} & 1 \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 2 & 2 \\ \mathrm{NIL} & 3 & \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} \\ 4 & \mathrm{NIL} & 4 & \mathrm{NIL} & \mathrm{NIL} \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 5 & \mathrm{NIL} \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(1)} = \begin{pmatrix} \mathrm{NIL} & 1 & 1 & \mathrm{NIL} & 1 \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 2 & 2 \\ \mathrm{NIL} & 3 & \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} \\ 4 & 1 & 4 & \mathrm{NIL} & 1 \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 5 & \mathrm{NIL} \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(2)} = \begin{pmatrix} \mathrm{NIL} & 1 & 1 & 2 & 1 \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 2 & 2 \\ \mathrm{NIL} & 3 & \mathrm{NIL} & 2 & 2 \\ 4 & 1 & 4 & \mathrm{NIL} & 1 \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 5 & \mathrm{NIL} \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(3)} = \begin{pmatrix} \mathrm{NIL} & 1 & 1 & 2 & 1 \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 2 & 2 \\ \mathrm{NIL} & 3 & \mathrm{NIL} & 2 & 2 \\ 4 & 3 & 4 & \mathrm{NIL} & 1 \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & 5 & \mathrm{NIL} \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad \Pi^{(4)} = \begin{pmatrix} \mathrm{NIL} & 1 & 4 & 2 & 1 \\ 4 & \mathrm{NIL} & 4 & 2 & 1 \\ 4 & 3 & \mathrm{NIL} & 2 & 1 \\ 4 & 3 & 4 & \mathrm{NIL} & 1 \\ 4 & 3 & 4 & 5 & \mathrm{NIL} \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad \Pi^{(5)} = \begin{pmatrix} \mathrm{NIL} & 3 & 4 & 5 & 1 \\ 4 & \mathrm{NIL} & 4 & 2 & 1 \\ 4 & 3 & \mathrm{NIL} & 2 & 1 \\ 4 & 3 & 4 & \mathrm{NIL} & 1 \\ 4 & 3 & 4 & 5 & \mathrm{NIL} \end{pmatrix}$$

9. Srautai tinkluose

Tarkime, kad $G = (V, E)$ yra jungus digrafas su išskirtomis šaltinio $s \in V$ ir tikslo $t \in V$ viršūnėmis. Be to, jame yra apibrėžta *talpos* funkcija $c : E \to \mathbf{R}^+$. Patogumui išplėskime jos apibrėžimo sritį iki $V \times V$, imdami $c(uv) = 0$, jei $uv \notin E$. Tokį digrafą vadinsime *tinklu*.

Apibrėžimas. *Srautu tinkle* $G = (V, E)$ *vadinsime funkciją* $f : V \times V \to \mathbf{R}$, *jei patenkinamos sąlygos:*

(i) $f(uv) \leq c(uv)$*;*
(ii) $f(uv) = -f(vu)$*;*
(iii) kiekvienai $u \in V \setminus \{s,t\}$ *turime*

$$\sum_{v \in V} f(uv) = 0.$$

Paskutinė sąlyga vadinama *Kirchhoff'o* aksioma. Dydį

$$|f| := \sum_{v \in V} f(sv) = \sum_{v \in V} f(vt)$$

vadinkime *srauto didumu.* Žargonu kalbant, tai ištekėjusio iš s srautas. Aksiomos (iii) dėka jis lygus įtekėjusio į t srautui.

Iš aksiomų išplaukia keletas paprastų savybių:

$$-c(vu) \leq f(uv) \leq c(uv),$$

$$f(uv) = 0, \text{jei } uv \notin E ir vu \notin E.$$

$$\sum_{u \in \Gamma^+(v)} f(uv) = \sum_{u \in \Gamma^-(v)} f(uv),$$

čia $\Gamma^+(v)$ ir $\Gamma^-(v)$ yra viršūnių, iš kurių patenkama į v ir į kurias patenkama iš v, aibės atitinkamai.

Viena iš pagrindinių problemų – *Maksimalaus srauto problema* formuluojama paprastai: rasti srautą f su maksimaliu didumu. Panašus uždavinys iškyla turint keletą šaltinių $s_1, \dots s_k$ ir keletą tikslo $t_1, \dots, t_r$ viršūnių. Tačiau šiuo atveju, įvedus papildomą šaltinį s, tikslo viršūnę t ir briaunas ss_i bei t_jt, be to, apibrėžus $c(ss_i) = \infty$ ir $c(t_jt) = \infty$, pakanka spręsti pirmąjį uždavinį.

Įveskime keletą patogių žymenų. Tegu

$$f(X,Y) = \sum_{x \in X, y \in Y} f(xy), \quad X, Y \subset V$$

ir

$$c((X,Y) = \sum_{x \in X, y \in Y} c(xy), \quad X, Y \subset V.$$

Pastebėkime:
1) $f(X,X) = 0$;
2) $f(X,Y) = -f(Y,X)$;
3) $f(X, Y \cup Z) = f(X,Y) + f(X,Z)$, jei $Y \cap Z = \emptyset$;
4) $|f| = f(V, \{t\}) =: f(V,t) = f(\{s\}, V) = f(s,V)$.

Nagrinėsime Ford'o ir Fulkerson'o 1956 metais pasiūlytą metodą maksimalaus srauto problemai spręsti. Jo idėja ieškoti srauto iteracijų, nuosekliai jį didinant. Pradedama nuo $f \equiv 0$ ir, radus $(s-t)$ taką p, jo tiesioginėse briaunose srautas yra didinamas, o briaunose, einančiose priešinga takui kryptimi, – mažinamas. Tokį padidinimą galima atlikti dydžiu

$$c(p) := \min\{c(uv) : \ uv \in p\}$$

vadinamu *tako talpa.* Kai $c(p) > 0$, pats takas vadinamas *auginančiu* srautą. Algoritme daug kartų iškviečiama procedūra:

F-F (G, s, t):

1. $f \leftarrow 0$
2. **while** egzistuoja auginantis takas p
3.   **do** "augink f take p"
4. **return** f

Realizuojant tokią idėją, patogu naudoti *likutiniais tinklais* $G_f = (V, E_f)$. Jo apibrėžimui panaudokime *likutinę talpą*

$$c_f(uv) = c(uv) - f(uv).$$

Ją galime suvokti kaip papildomą talpą, jei srautas $f(uv)$ šia briaunoje buvo "praleistas". Kai $f(uv) < 0$, vietoje nulinės talpos briaunos atsirado briauna su teigiama talpa. Todėl likutiniame grafe briaunų aibė

$$E_f = \{uv \in V \times V : \ c_f(uv) > 0\}.$$

Srautų f_1 ir f_2 sumą apibrėžkime kaip pataškinę funkcijų sumą.

1 lema. *Jei f yra srautas tinkle $G = (V, E)$, $G_f(V, E_f)$ – likutinis tinklas su srautu f', tai $f + f'$ – srautas pradiniame tinkle G. Be to, $|f + f'| = |f| + |f'|$.*

Įrodymas. Patikrinti (i)-(iii) aksiomas. ◇

2 lema. *Tegu f yra srautas tinkle $G = (V, E)$ ir p – (s-t) takas. Apibrėžkime*

$$f_p(uv) := \begin{cases} c_f(p), & \textit{jei } uv \in p, \\ -c_f(p), & \textit{jei } vu \in p, \\ 0 & \textit{likusiais atvejais.} \end{cases}$$

Tada f_p yra srautas likutiniame tinkle G_f ir $|f_p| = c_f(p)$.

Įrodymas. Patikrinti (i)-(iii) aksiomas. ◇

Išvada. *Radę auginantį taką p, galime apibrėžti srautą $f + f_p$, kurio didumas yra $|f| + c_f(p)$.*

Algoritmai naudos ir tinklo pjūvius $V = S \cup T$ su savybe $s \in S$ ir $t \in T$. Dydis $c(S, T)$ bus vadinamas *pjūvio talpa*, o $f(S, T)$ – srautu, *tekančiu iš S į T.*

3 lema. *Bet kokiam pjūviui $V = S \cup T$ turime $|f| = f(S, T)$.*

Įrodymas. Galime pasinaudoti 1)-5) savybėmis:

$$f(S,T) = f(S,V) - f(S,S) = f(S,V) = f(s,V) + f(S-s,V) = f(s,V) = |f|.$$

Išvada. $|f| \le c(S,T)$.

Dabar galime pateikti svarbiausią teoremą apie srautus tinkluose.

Maksimalaus srauto ir minimalaus pjūvio teorema (Fordo-Fulkersono t.). *Jei f yra srautas tinkle $G = (V,E)$ su šaltiniu s ir tikslo viršūne t, tai šie teiginiai yra ekvivalentūs:*

(i) *f yra maksimalus srautas;*

(ii) *likutinis tinklas G_f neturi auginančių takų;*

(iii) *kažkokiam pjūviui (S,T) turime $|f| = c(S,T)$.*

Įrodymas. $(i) \Rightarrow (ii)$. Jei būtų priešingai, t.y. G_f turėtų auginantį taką, nors f būtų maksimalus srautas, tada pagal 2 lemos išvadą galėtume srautą padidinti.

$(ii) \Rightarrow (iii)$. Tarkime, kad G_f neturi auginančio $s-t$ tako. Apibrėžkime

$$S = \{v \in V : \exists\ s - v \text{ takas likutiniame tinkle } G_f\}$$

ir $T = V \setminus S$. Skaidinys $V = S \cup T$ yra pjūvis. Pakanka pastebėti, kad $s \in S$, o $t \in T$. Be to, kiekvienai porai $u \in S$ ir $v \in T$ turime $f(uv) = c(uv)$. Priešingu atveju nelygybė $c(uv) > f(uv)$ rodytų briaunos uv likutiniame grafe egzistavimą, be to, tada $v \in S$. Tai prieštara. Vadinasi, pagal 3 lemą $|f| = f(S,T) = c(S,T)$.

$(iii) \Rightarrow (i)$. Pagal 3 lemos išvadą visiems pjūviams $|f| \le c(S,T)$. Aišku, kad lygybė galima tik $|f|$ pasiekus maksimumą. ◇

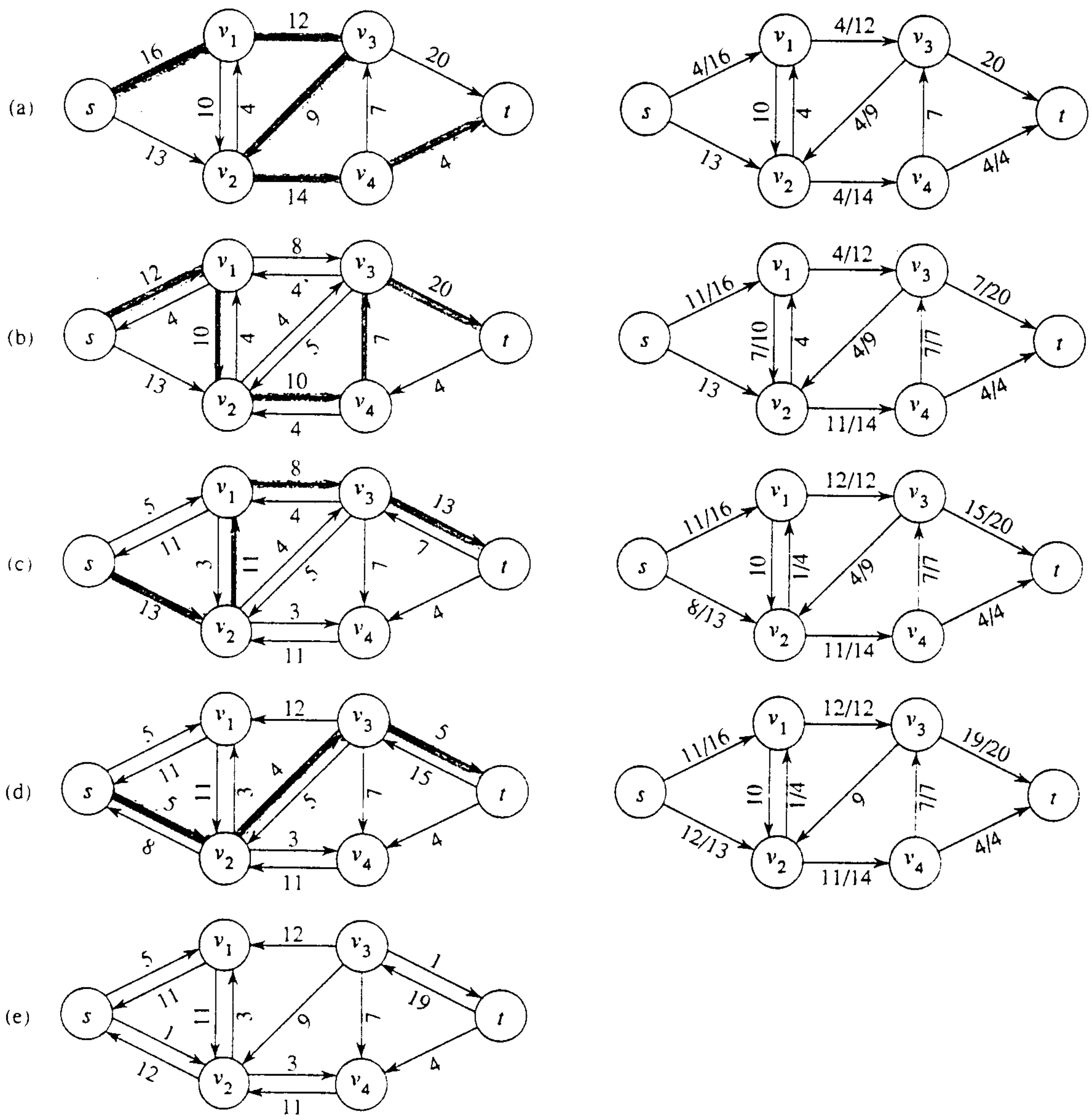
10. Fordo-Fulkersono metodas ir Edmonds-Karpo algoritmas

Realizuokime 9 skyrelyje aprašytą idėją, kaip gauti maksimalųjį srautą. Pradžioje susitarkime žymėti simboliu $f[uv]$ srautą briaunoje, jei šis simbolis programoje reiškia kintamąjį. Jis darant iteracijas vis atnaujinamas. Fordo-Fulkersono metodą, viršuje pažymėtą kaip *F-F*(G,s,t), galėtume realizuoti tokia operacijų seka.

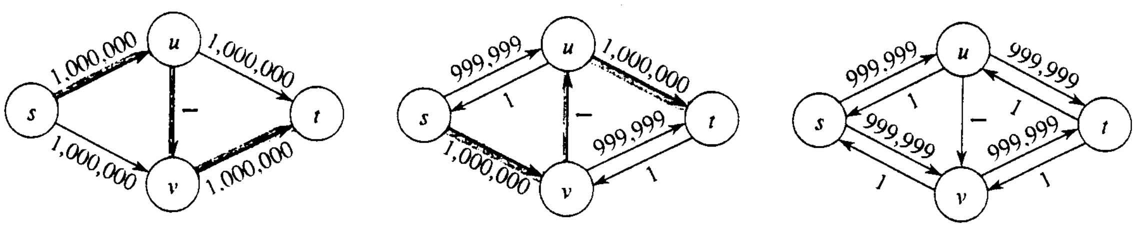
F-F-met (G,s,t):

```
for ∀ uv ∈ E
    do f[uv] ← 0
       f[vu] ← 0
while egzistuoja auginantis takas p
    do c_f(p) ← min{c_f(uv) : uv ∈ p}
       for ∀ uv ∈ p
           do f[uv] ← f[uv] + c_f(p)
              f[vu] ← −f[uv]
END
```

Išnagrinėti pavyzdį:



Jei 4-ame žingsnyje auginantis takas ieškomas nevykusiai, tai srauto auginimas labai lėtas. Žr. pavyzdį:



(a) (b) (c)

Pavyzdys rodo, kad procedūros trukmė priklauso net nuo srauto didumo. Iš tiesų, kiekviena iteracija sugebėdavo jo didumą pakelti vienetu. Be to, yra tokių pavyzdžių (!), kad iteracijos iš viso nekonverguos į kažkokią ribą. Tai gali būti susiję su briaunų talpų nebendramatiškumu. Vis tik sveikareikšmių talpų atveju **F-F-met**(G, s, t) yra baigtinė procedūra, kurios trukmė $O(|f^*||E|)$, čia f^* – maksimalusis srautas.

Reikiamus metodo pataisymus atskleidė Edmonds'as ir Karp'as. Jų esmė – 4 žingsnyje ieškomas ne bet koks, o *trumpiausias* auginantis takas. Šis takas turi mažiausią skaičių briaunų, ir eina briaunų kryptimi. Čia patogu kiekvieną kartą pritaikyti **P-Plot**(G) algoritmą.

Tegu, kaip ir anksčiau $\delta(u, v)$ yra trumpiausio $u \rightsquigarrow v$ tako ilgis (braunų skaičius jame). Algoritmo analizei ir korektiškumui įrodyti pradžioje pateiksime keletą lemų.

1 lema. *Jei tinkle $G = (V, E)$ su šaltinio ir tikslo viršūnėmis s ir t bei talpos funkcija $c(uv)$ vykdomas Edmondso-Karpo algoritmas, likutiniuose tinkluose, padidėjus srautui, atstumai $\delta_f(s, v)$, $v \in V \setminus \{s, t\}$, monotoniškai didėja plačiaja prasme.*

Įrodymas. Tarkime, kad f yra srautas prieš jo padidinimą kažkokiame take, o f' - padidintas srautas. Tegu priešingai, kažkokiai viršūnei v

$$\delta_{f'}(s, v) < \delta_f(s, v).$$

Iš visų tokių v išrinkime arčiausią nuo s tinkle $G_{f'}(V, E_{f'})$. Todėl iš visų $u \in V \setminus \{s, t\}$ su savybe $\delta_{f'}(s, u) < \delta_f(s, u)$ viršūnė v tenkins dar ir

$$\delta_{f'}(s, v) < \delta_{f'}(s, u).$$

Formaliai užrašant,

$$\big\{\delta_{f'}(s, u) < \delta_f(s, u)\big\} \;\Rightarrow\; \big\{\delta_{f'}(s, v) \le \delta_{f'}(s, u)\big\}.$$

Tai ekvivalentu implikacijai

$$\big\{\delta_{f'}(s, v) > \delta_{f'}(s, u)\big\} \;\Rightarrow\; \big\{\delta_{f'}(s, u) \ge \delta_f(s, u)\big\}. \tag{1}$$

Tinkle $G_{f'}$ imkime trumpiausią $s \rightsquigarrow v$ taką p' pavidalo

$$s \rightsquigarrow u \to v.$$

Žinom, kad

$$\delta_{f'}(s,u) = \delta_{f'}(s,v) - 1 < \delta_{f'}(s,v).$$

Iš (1) išplaukia, kad

$$\delta_f(s,u) \le \delta_{f'}(s,u). \tag{2}$$

Dabar grįžtame į tinklą G_f. Tegu u ir v yra jau anksčiau aptartos viršūnės. Jei

$$f[uv] < c(uv),$$

tai $uv \in E_f$ ir pagal (2)

$$\delta_f(s,v) \le \delta_f(s,u) + 1 \le \delta_{f'}(s,u) + 1 = \delta_{f'}(s,v).$$

Tai rodo, kad po srauto auginimo $\delta_{f'}(s,v)$ nesumažėjo. Prieštara įrodo, kad

$$f[uv] = c(uv),$$

bet tada $uv \notin E_f$. Vadinasi, auginantis $s \rightsquigarrow t$ takas turi turėti briauną vu. Jo pavidalas yra

$$s \rightsquigarrow v \to u \rightsquigarrow t.$$

Edmondso-Karpo algoritme jis yra trumpiausias, jo dalis $s \rightsquigarrow v \to u$ irgi yra trumpiausias takas tarp tarpinių viršūnių. Vėl pasinaudoję (2), gauname

$$\delta_f(s,v) = \delta_f(s,u) - 1 \le \delta_{f'}(s,u) - 1 = \delta_{f'}(s,v) - 2 < \delta_{f'}(s,v).$$

Prieštara įrodo lemos tvirtinimą. $\diamond$

Apibrėžimas. *Tako $p:\ s \rightsquigarrow t$ briauną $uv \in E_f$, tenkinančią sąlygą*

$$c_f(p) = c_f(uv),$$

vadinsime kritine.

2 lema. *Ta pati briauna, vykdant Edmondso-Karpo algoritmą, gali būti kritine ne daugiau kaip*

$$|V|/2 - 1$$

kartų.

Įrodymas. Atlikus srauto padidinimą take, kritinė briauna likutiniame grafe išnyksta. Ji yra trumpiausiame $s \rightsquigarrow t$ take. Todėl

$$\delta_f(s,v) = \delta_f(s,u) + 1.$$

Kada ji vėl pasirodo kokiame nors likutinio tinklo auginančiame take? Tai gali atsitikti tik prieš tai pasirodžius vu. Lemos 1 įrodyme pastebėjome, kad šios briaunos pasirodymo metu auginančiame take ir esant srautui f', turime

$$\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1 \geq \delta_f(s,v) + 1 = \delta_f(s,u) + 2.$$

Vadinasi, briauna uv vėl taps kritine, kai atstumas $\delta_f(s,u)$ padidės bent 2. Kadangi $\delta_f(s,u) \leq |V| - 2$, iš čia gauname atsakymą. Lema įrodyta. $\diamond$

Išvada. *Per visą Edmondso-Karpo algoritmo vykdymo laikotarpį yra ne daugiau kaip $|E||V|/2$ kritinių briaunų su galimais pasikartojimais.*

Teorema. *Edmondso-Karpo algoritmas duoda maksimalų srautą per $O(|V||E|^2)$ žingsnių.*

Įrodymas. Išnykus kritinėms briaunoms nėra ir auginančių $s \rightsquigarrow t$ takų. Pagal Fordo-Fulkersono teoremą tada f jau yra maksimalusis srautas.

Pagal ką tik padarytą išvadą kritinių briaunų skaičius yra $O(|V||E|)$, o kiekvieno auginančio tako ilgis neviršija $|E|$. Jo briaunoms vis reikia perskaičiuoti talpas ir liekamuosius srautus. Tai užtrunka $O(|E|)$ laiko. Tad algoritmo realizacijos trukmė yra $O(|V||E|^2)$. $\diamond$

11. Maksimalusis dvidalis suporavimas

Suporavimu grafe $G = (V, E)$ vadiname nepriklausomų briaunų poaibį. Kitaip tariant, $M \subset E$ yra suporavimas, jei kiekviena viršūnė $v \in V$ turi ne daugiau kaip vieną incidenčią jai briauną iš M. *Maksimalusis* suporavimas turi didžiausią skaičių briaunų. Mes nagrinėsime maksimalaus suporavimo uždavinį dvidaliame grafe, todėl prie termino pridėsime žodį „dvidalis" . Tegu jame $V = L \cup R$ yra viršūnių aibės skaidinys.

Apibrėžimas. *Visiškasis suporavimas dvidaliame grafe $G = (L \cup R, E)$ yra toks poravimas, kai visos vienos dalies viršūnės yra incidenčios suporavimo briaunoms.*

Prisiminkite Hallo vedybų problemą! Aišku, kad visiškojo suporavimo M atveju $|M| = \min\{|L|, |R|\}$. Toks suporavimas ne visada egzistuoja, bet visada galime rasti maksimąlųjį dvidalį suporavimą.

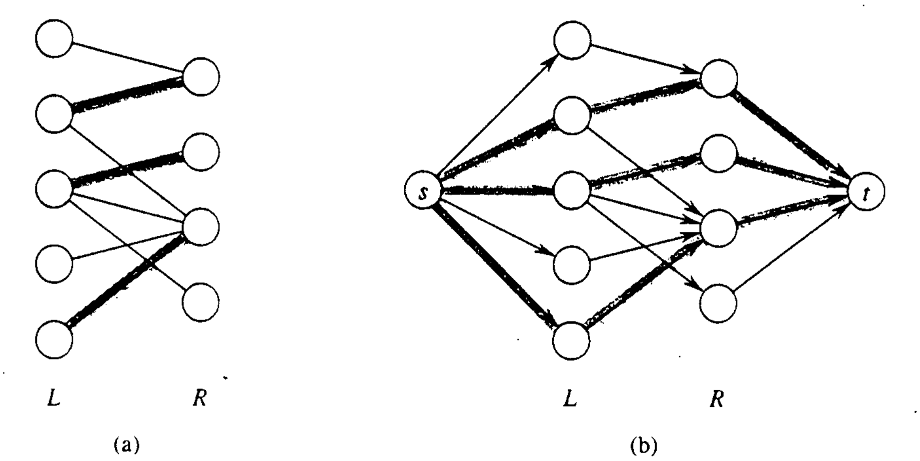
Pritaikysime Fordo-Fulkersono algoritmą. Tuo tikslu sudarome asocijuotą srauto tinklą $G' = (V', E')$ su

$$V' = L \cup R \cup \{s,t\}, \qquad E' = \{uv \in E\} \cup \{su : \; u \in L\} \cup \{vt : \; v \in R\}$$

ir visoms briaunoms suteikiame kryptis ir vienetines talpas.

1 lema. *Jei M yra dvidalis suporavimas, tai asocijuotame tinkle egzistuoja srautas f, kurio didumas yra $|f| = |M|$. Atvirkščiai, turėdami tinkle G' sveikareikšmį srautą f, turime ir suporavimą M, kurio didumas yra $|M| = |f|$.*

Įrodymas. Išnagrinėkime paveikslą:



Tarkime, kad M yra dvidalis suporavimas ir apibrėžkime funkciją $f : E \to \mathbf{Z}$

$$f(su) = f(uv) = f(vt) = 1$$

ir

$$f(us) = f(vu) = f(tv) = -1,$$

jei $uv \in M$, bei $f(uv) = 0$, jei $(u,v) \in V' \times V' \setminus E'$. Ji tenkina srauto aksiomas. Matome, kad takai

$$s \to u \to v \to t,$$

kai $uv \in M$ yra nepriklausomi. Todėl jais galime ,,praleisti" vienetinio didumo srautus. Tai apibrėžia tinklo G' srautą, kurio didumas lygus pjūvio

$$(\{s\} \cup L) \cup (R \cup \{t\})$$

talpai $|M|$.

Atvirkščiai, Jei turime srautą f su reikšmėmis $\mathbf{Z}$. Pažymėkime

$$M = \{uv : \ u \in L, \ v \in R, \ f(uv) > 0\}.$$

Kadangi $c(uv) = 1$, tai visosms šioms briaunoms $f(uv) = 1$. Ar jos yra nepriklausomos? Jeigu būtų priešingai, t.y. kažkokia viršūnė būtų incidenti dviems briaunoms iš M. Vadinasi, joje būtų pažeidžiamas Kirchhofo dėsnis. Vadinasi, M yra suporavimas. Raskime jo galią $|M|$.

Pastebėkime, kad $|M| = f(L,R)$, todėl

$$|M| = f(L,V') - f(L,L) - f(L,s) - f(L,t) = 0 - 0 + f(s,L) - 0 = f(s,V') = |f|.$$

Lema įrodyta. $\diamond$

2 lema. *Jei tinkle talpa yra sveikareikšmė, tai maksimalus srauto didumas yra sveikareikšmis.*

Įrodymas. Praktiškai realizuokime Fordo-Fulkersono procedūrą. Kiekviename žingsnyje srautai iteruojami pagal taisyklę $f[uv] \leftarrow f[uv] + c_f(p)$, čia $c_f(p)$ yra sveikareikšmė likutinė talpa take p. Pritaikę indukciją pagal iteracijų skaičių, įrodome teiginį. $\diamond$

1 teorema. *Maksimalaus suporavimo galia lygi maksimalaus srauto asocijuotame tinkle didumui.*

Įrodymas. Tai yra išvada iš 1 ir 2 lemų bei Fordo-Fulkersono teoremos. $\diamond$

Aprašytas algoritmas palengvina ir visiškojo suporavimo kriterijaus (Hallo sąlygos) patikrinimą. Tą demonstrusime reguliarių dvidalių grafų atveju. Primename, kad d reguliarus grafas turi visas d laipsnio viršūnes.

2 teorema. *Dvidalis $d \geq 1$ reguliarus grafas turi visiškąjį suporavimą.*

Įrodymas. Imkime asocijuotą tinklą. Tegu $|L| \leq |R|$. Jo minimalaus pjūvio talpa lygi $|L|$. Jame maksimalusis srautas turi didumą $|L|$. Vadinasi, pagal 1 teoremą yra tokios galios suporavimas, kuris suporuoja visas L viršūnes. $\diamond$

12. Priešsraučio stūmimo algoritmas

Šiuo algoritmu irgi yra sprendžiamas maksimalaus srauto uždavinys. Yra pasiekiamas $O(|V|^3)$ vykdymo trukmės laikas. Esminis skirtumas nuo ankstesnių metodų yra tas, kad atsisakoma auginančių takų ieškojimo, tarpiniuose žingsniuose atsisakoma net Kirchhofo aksiomos.

Apibrėžimas. *Priešsraučiu tinkle vadinama funkcija $F : V \times V \to \mathbf{R}$, tenkinanti sąlygas:*

(i) $f(uv) = -f(vu)$;
(ii) $f(uv) \leq c(uv)$;
(iii) $f(V,u) = \sum_{v \in V} f(vu) \geq 0$ *kiekvienai* $u \in V \setminus \{s\}$.

Dydį $e(u) := f(V,u)$ vadinsime *pertekliniu srautu* viršūnėjė u. Viršūnę $u \in V \setminus \{s,t\}$ su $e(u) > 0$ vadinsime *pertekline*. Galima įsivaizduoti, kad viršūnės turi neribotus rezervuarus, kuriuose laikinai patalpinamas perteklinis srautas $e(u)$. Šis įvaizdis dar patogus ir tuo, kad algoritmo vykdymo metu viršūnė bus ,,keliama" aukštyn, kad šis perteklius galėtų ,, ištekėti" į gretimą viršūnę. Kaip ir anksčiau skaičiuosime likutinius tinklus $G_f = (V, E_f)$, bet f jau bus priešsrautis ir nebūtinai srautas.

Apibrėžimas. *Aukščiu vadinama funkcija $h : V \to \mathbf{N}$, tankinanti sąlygas:*
(i) $h(s) = |V|$, $h(t) = 0$;
(ii) $h(u) \leq h(v) + 1$, *jei* $uv \in E_f$.

Vadinasi, jei f yra priešsrautis tinkle $G = (V, E)$ ir h – aukščio funkcija, tai poroms $(u,v) \in V \times V$ iš

$$h(u) > h(v) + 1$$

išplauks $uv \not\in E_f$.

Viena iš pagrindinių operacijų yra **Stumk**(uv). Ji bus vykdoma, kai

$$c_f(uv) > 0, \qquad h(u) = h(v) + 1. \tag{1}$$

Jos metu briauna uv išstumiamas srautas

$$d_f(uv) = \min\{e(u),\, c_f(uv)\}.$$

Stumk(uv):

1. $d_f(uv) \leftarrow \min\{e[u],\, c_f(uv)\}$
2. $f[uv] \leftarrow f[uv] + d_f(uv)$
3. $f[vu] \leftarrow -f[uv]$
4. $e[u] \leftarrow e[u] - d_f(uv)$
5. $e[v] \leftarrow e[v] + d_f(uv)$

Pastebėkime, kad ši operacija nepriklauso nuo aukščio, bet vykdant (1) sąlyga yra būtina. Kaip anksčiau buvome pastebėję, likutiniame tinkle briaunos uv ir nebus, jei $h(u) > h(v)+1$. Stūmimas yra *prisotinantis*, jei $d_f(uv) = c_f(uv)$, todėl po jo likutiniame tinkle ši briauna išnyks. Priešingu atveju, stūmimas yra *neprisotinantis*.

Kita svarbi operacija yra **Kelk**(u), atliekama, jei viršūnė yra perteklinė ir visoms briaunoms $uv \in E_f$ su $c_f(uv) > 0$, turime $h(u) \leq h(v)$. Paskutinė sąlyga reiškia, kad iš u yra ne aukščiau, nei kitos jos kaimynės į kurias patenkama likutiniame tinkle. Primename, kad s ir t nelaikomos pertekliėmis, tad jos nebus keliamos.

Kelk(u):

1. $h[u] \leftarrow 1 + \min\{h[v] :\ uv \in E_f\}$

Pastebėkime, kad perteklinei viršūnei visada $e(u) = f(V, u) > 0$, todėl yra tokia viena $v \in V$, kad $f(vu) > 0$. Tada

$$c_f(uv) = c(uv) - f(uv) = c(uv) + f(vu) > 0.$$

Taigi, egzistuoja $uv \in E_f$.

Dabar galime aprašyti bendrus priešsraučių algoritmų principus. Pradėkime nuo inicializacijos.

INIT(G, s):

1. **for** $\forall u \in V$
2.   **do** $h[u] \leftarrow 0$
3. $e[u] \leftarrow 0$
4. **for** $\forall uv \in E$
5.   **do** $f[uv] \leftarrow 0$
6. $f[vu] \leftarrow 0$
7. $h[s] \leftarrow |V|$
8. **for** $\forall u \in Adj[s]$
9.   **do** $f[su] \leftarrow c(su)$

10. $\qquad f[us] \leftarrow -c(su)$
11. $\qquad e[u] \leftarrow c(su)$

END

Taigi **INIT**(G, s) sukuria priešsrautį

$$f[uv] = \begin{cases} c(uv), \text{ jei u=s,} \\ -c(vu), \text{ jei v=s,} \\ 0 \text{ likusiais atvejais.} \end{cases}$$

Visos viršūnės, gretimos su s gavo srauto perteklių. Tuo pačiu apibrėžėme ir aukščio funkcijos pirmąją iteraciją. Įsitikinkite, kad visos aukščio funkcijos sąlygos yra patenkintos!

Bendra Pr-St procedūra:

1. INIT(G, s)
2. **while** egzistuoja galimybė naudoti *Kelk*(u) ar *Stumkuv*)
3. $\qquad$ **do** tai.

1 lema. *Tarkime, kad f yra priešsrautis tinkle $G = (V, E)$ su šaltinio viršūne s ir tikslo viršūne t bei aukščio funkcija h. Jei u yra perteklinė viršūnė, tai galime taikyti* **Kelk**(u) *arba* **Stumk**(uv) *operaciją.*

Įrodymas. Kiekvienai likutinio grafo briaunai turime

$$h(u) \leq h(v) + 1.$$

Jei **Stumk**(uv) negalime taikyti, tai turi būti $h(u) < h(v) + 1$. Taigi, $h(u) \leq h(v)$ ir galime taikyti **Kelk**(u). $\diamond$

2 lema (Aukščio funkcijos savybės). *Atliekant* **Bendrą Pr-St** *procedūrą, $h[u]$ nemažėja. Jei u bent kartą kėlėme, tai jos aukštis padidėjo bent vienetu. Visada $h[u]$ tenkina aukščio funkcijos reikalavimus.*

Įrodymas. Stūmimo metu aukščio funkcija nekeičiama. Reikia patikrinti paskutinį lemos teiginį. Pritaikome indukciją pagal kėlimų skaičių. Po inicializacijos h tenkino minėtus reikalavimus.

Tegu $uv \in E_f$. Jei šiame likutiniame tinkle u kėlėme, tai prieš tai buvo $h[u] \leq h[v]$, o po pakėlimo – $h(u) \leq h(v) + 1$. Vadinasi, aukščio funkcijos sąlyga galioja ir dabar.

Tegu $wu \in E_f$. Jei šiame likutiniame tinkle u kėlėme, tai prieš tai galiojo indukcijos aksioma ir buvo $h[w] \leq h[u] + 1$, o po pakėlimo – $h(w) \leq h(u) + 1$. Vadinasi, aukščio funkcijos sąlyga irgi dabar yra patenkinta. $\diamond$

Dabar pastebėsime vieną likutinio tinklo savybę, kurios nebuvo naudojat srautus, o ne priešsraučius.

3 lema. *Tarkime, kad f yra priešsrautis tinkle $G = (V, E)$ su šaltinio viršūne s ir tikslo viršūne t bei aukščio funkcija h. Likutiniame tinkle $G_f = (V, E_f)$ nėra $s - t$ tako, einančio briaunų kryptimis.*

Įrodymas. Tegu $p = v_0 v_1 \cdots v_k$ yra toks takas, $s - v_0$, $t = v_k$ ir $k < |V|$. Jo briaunos $v_i v_{i+1} \in E_f$. Be to, $h(v_i) \leq h(v_{i+1}) + 1$, kai $i = 0, \ldots, k-1$. Vadinasi,

$$h(s) \leq h(t) + k = k < |V|.$$

Bet jau inicializacijoje turėjome $h(s) = |V|$. Prieštara įrodo lemos teiginį. $\diamond$

Teorema. *Atlikus* **Bendrą Pr-St** *procedūrą, priešsrautis yra maksimalusis srautas.*

Įrodymas. Pabaigoje bet kokia viršūnė $u \in V \setminus \{s, t\}$ negali būti perteklinė. Visą procedūros vykdymo laiką f buvo priešsraučiu likutiniuose grafuose. Kai nėra perteklinių viršūnių, jis yra srautas. Aukščio funkcija irgi išlieka visą laiką apibrėžta. Pagal 3 lemą pabaigoje nebėra $s - t$ takų. Vadinasi, srauto padidinti nebėra kaip. Jis yra maksimalus. $\diamond$

13. Priešsraučio stūmimo algoritmo analizė

Dabar įsitikinsime, kad šiuo algoritmu maksimaliojo srauto problema išsprendžiama per $O(|V|^2|E|)$ laiko vienetų arba žingsnių. Pradžioje pateiksime keletą lemų. s

1 lema. *Tarkime, kad f yra priešsrautis tinkle $G = (V, E)$ su šaltinio viršūne s ir tikslo viršūne t. Jei u yra perteklinė viršūnė, tai likutiniame tinkle G_f yra $u - s$ takas.*

Įrodymas. Tuoj po inicializacijos tai trivialu. Sudarykime viršūnių aibės skaidinį. Tegu

$$U = \{v : \exists (u - v) \text{ takas, priklausantis } G_f\}$$

ir tarkime, kad $s \notin U$. Pažymėkime $\bar{U} = V \setminus U$. Taigi, U sudaro visos pasiekiamos iš u viršūnės, o $\bar{U}$ - nepasiekiamos. Tegu v yra pasiekiama, o w – nepasiekiama, įrodysime, kad

$$f(wv) \leq 0. \tag{1}$$

Priešingas (1)-am teiginys rodo, kad $f(vw) < 0$, Vadinasi, likutinė talpa

$$c_f(vw) = c(vw) - f(vw) > 0.$$

Todėl egzistuoja briauna $vw \in E_f$. Bet tada takas $u \rightsquigarrow v \to w$ rodo, kad grafe G_f viršūnė w yra pasiekiama. Prieštara įrodo (1) nelygybę. Iš čia gauname

$$f(U, \bar{U}) \leq 0.$$

Todėl

$$0 < e(U) = \sum_{u \in U} e(u) = f(V, U) = f(\bar{U}, U) + f(U, U) = f(\bar{U}, U) \leq 0.$$

Prieštara baigia įrodyti lemos teiginį. $\diamond$

2 lema. *Tarkime, kad tinkle $G = (V, E)$ su šaltinio viršūne s ir tikslo viršūne t yra apibrėžta aukščio funkcija h. Vykdant* **Bendrą Pr-St** *procedūrą, visada ir visoms $u \in V$*

$$h[u] \leq 2|V| - 1.$$

Įrodymas. Visada nekintamai turime $h(s) = |V|$ ir $h(t) = 0$. Tegu u yra bet kokia kita viršūnė. Imkime 1 lemoje nurodytą $u \rightsquigarrow s$ taką

$$p = v_0 v_1 \dots v_k, \qquad v_0 = u, \quad v_k = s, \quad k \leq |V| - 1.$$

Kadangi $v_i v_{i+1} \in E_f$, tai $h[v_i] \leq h[v_{i+1}] + 1$. Sudėję pagal i, gauname

$$h[v_0] = h[u] \leq h[v_k] + k = h[s] + k \leq 2|V| - 1.$$

Lema įrodyta. ◇

Išvada. *Vykdant* **Bendrą Pr-St** *procedūrą, atliekama ne daugiau negu* $2|V|^2$ *viršūnių kėlimų.*

Įrodymas. Kiekvieną kartą bet kokia viršūnė keliama bent per vienetą. ◇

Dabar rasime priešsraučio stūmimų skaičių. Pradėkime nuo prisotinančiųjų. Prieš jį vykdant turi galioti sąlyga $c(uv) \leq e[u]$, o po jo – $c_f(uv) = 0$.

3 lema. *Vykdant* **Bendrą Pr-St** *procedūrą, atliekama ne daugiau negu* $2|V||E|$ *prisotinančių stūmimų.*

Įrodymas. Imkime porą $u, v \in V$. Stūmimai gali būti vykdomi arba uv arba briauna vu. Tegu jau įvykdėm pirmąjį prisotinantį stūmimą, t.y., jį vykdėme briauna uv. Turėjo būti $e[u] \geq c(uv)$. Kadangi likutinė talpa $c_f(uv) = 0$, briauna uv dingo. Iki kito stūmimo ta pačia briauna turėjo būti atliktas stūmimas briauna vu. O tam reikėjo v pakelti bent du kartus. Panašiai, būtų tekę elgtis ir su briauna vu.

Nagrinėkime seką

$$h[u] + h[v],$$

kai tarp briaunų u ir v, viena ar kita kryptimi, yra vykdomi prisotinantys stūmimai. Turi būti

$$h[u] + h[v] \geq 1.$$

Pagal 2 lemą po paskutinio stūmimo

$$h[u] + h[v] \leq (2|V| - 1) + 2(|V| - 2) = 4|V| - 3.$$

Kadangi tarp stūmimų yra kėlimas bent per du, tai stūmimų skaičius neviršys

$$(4|V| - 3)/2 + 1 = 2|V| - 1$$

Padauginę šį įvertį iš briaunų skaičiaus, gauname norimą rezultatą. ◇

4 lema. *Vykdant* **Bendrą Pr-St** *procedūrą, atliekama ne daugiau negu* $4|V|^2(|V| + |E|)$ *neprisotinančių stūmimų.*

Įrodymas. Dabar nagrinėkime funckiją

$$\Phi(X) := \sum_{v \in X} h[v];$$

čia X yra isų perteklinių viršūnių aibė. Po inicializacijos turime $\Phi(X) = 0$. Kėlimai nekeičia aibės X, o pagal 2 lemą vieną viršūnę galime iškelti į ne didesnį kaip $2|V|$ aukštį. Pagal išvadą iš viso, gal būt, pakartojamų viršūnių kėlimų yra ne daugiau kaip $2|V|^2$. Prisotinantys stūmimai iš u į v nekeičia aukščio (jis neviršija $2|V|$), bet po jo viršūnė v gali jau patekti į aibę X. Todėl pagal 3 lemą galimas funkcijos $\Phi(X)$ padidėjimas dėl šios priežasties irgi ne didesnis už $2|V| \times 2|V||E|$.

Neprisotinantis stūmimas iš u į v, priešingai, sumažina šią funkciją bent per vienetą. Iš tiesų, po jo u nebepriklauso X aibei, todėl $h(u)$ atsiėmė iš šios sumos, o net jei v naujai įsijungė į aibę X reikšmė $h(v) = h(u) - 1$ negalėjo vienetu kompensuoti nuostuolio. Taigi, pirmais dviem atvejais $\Phi(X)$ galėjo padidėti ne daugiau kaip iki

$$4|V|^2(|V| + |E|), \tag{2}$$

o neprisotinantys stūmimai mažina bent per vienetą, bet nuo to $\Phi(X)$ neigiamas negali pasidaryti. Vadinasi, jų skaičius neviršija (2) skaičiaus.

Lema įrodyta. ◇

Teorema. *Vykdant* **Bendrą Pr-St** *procedūrą, pakanka* $O(|V|^2)$ *bazinių* **Kelk** *ir* $O(|V|^2|E|)$ **Stumk** *operacijų. Visas algoritmo realizacijos laikas yra* $O(|V|^4)$.

Įrodymas. Pirmas teiginys išplaukia iš 2 lemos išvados, 3 ir 4 lemų. ◇

Kelk(u) metu reikia atlikti

$$h[u] \leftarrow 1 + \min\{h[v] : \ uv \in E_f\}.$$

Tai užima $O(|V|)$ laiko. Vadinasi, kėlimų užimamas laikas neviršija teoremoje nurodyto rėžio.

Vieman stūmimui pakanka $O(1)$ laiko, o visiems – $O(|V|^2|E|)$ laiko. ◇

14. „Kelk priekin" algoritmas

Priešsraučio stūmimo algoritme pagrindinės *Kelk* ir *Stumk* operacijos taikomos be jokios išankstinės tvarkos, kada tik jas galima taikyti. Reguliuojant šį procesą, algoritmą galima patobulinti ir vykdymo greitį padidinti nuo $O(|V|^2|E|)$ iki $O(|V|^3)$. Retiems tinklams tai yra esminis pagerinimas.

Naujas algoritmas vykdymo metu išsaugo viršūnių sąrašą. Pradėdamas nuo pradinės viršūnės iš eilės, jas, jeigu reikia, pakelia ir iškrauna išstumdamas perteklinį priešsrautį. Žinoma, sekantys kėlimai ir stūmimai, gali ją vėl padaryti pertekline, todėl panašias procedūras tenka kartoti keletą kartų. Kiekvieną kartą pakėlus viršūnę, ji įrašoma sąrašo priekyje, todėl ir algoritmas vadinamas „kelk priekin" vardu.

Tinklo $G = (V, E)$ su šaltinio viršūne s ir tikslo viršūne t bei priešsraučiu f briauna uv vadinama *tinkama* (arba leistina), jeigu

$$c_f(uv) > 0, \qquad h(u) = h(v) + 1.$$

Prisiminkime, kad anksčiau išnagrinėtame priešsraučio stūmimo algoritme, jei būdavo $e(u) > 0$, tai būtent tinkamomis briaunomis mes vykdydavome operaciją *Stumk*(uv). Tinklas $G_{f,h} = (V, E_{f,h})$ irgi vadinamas *tinkamu*, čia $E_{f,h}$ – visų tinkamų briaunų aibė.

1 lema. *Tegu $G = (V, E)$ – tinklas su priešsraučiu f ir aukščio funkcija h. Tinkamas tinklas $G_{f,h} = (V, E_{f,h})$ yra beciklis.*

Įrodymas. Tegu priešingai $p = v_0 v_1 \ldots v_k$, $v_0 = v_k$, o $k > 0$, – ciklas. Jo briaunos yra tinkamos, todėl

$$h(u_{i-1}) = h(u_i) + 1, \quad i = 1, \ldots, k.$$

Sudėję šias lygybes, gauname

$$\sum_{i=1}^{k} h(u_{i-1}) = \sum_{i=1}^{k} h(u_i) + k.$$

Aukščių sumos, esančios kairėje ir dešinėje, yra lygios. Todėl gauname prieštarą: $0 < k = 0$. ◇

2 lema. *Tegu tinkle $G = (V, E)$ su priešsraučiu f ir aukčio funkcija h viršūnė u yra perteklinė, o briauna uv – tinkama.* **Stumk**(uv) *operacija nesukuria naujų tinkamų briaunų, bet uv gali tapti netinkama.*

Įrodymas. Atlikus minimą operaciją, gali atsirasti briauna vu. Bet $h(v) = h(u) - 1$, todėl ji nebus tinkama. Prisotinančio stūmimo atveju briauna uv išnyks. ◇

3 lema. *Tegu tinkle $G = (V, E)$ su priešsraučiu f ir aukčio funkcija h viršūnė u yra perteklinė, bet nėra tinkamų briaunų, išeinančių iš u. Tada galima taikyti* **Kelk**(u) *operaciją. Po jos atsiranda bent viena tinkama briauna, išeinanti iš u, bet nebus į ją įeinančių tinkamų briaunų.*

Įrodymas. Neturėdami tinkamų briaunų, perteklinę viršūnę galime tik kelti. Po jos

$$h[u] = 1 + \min\{h[v] : \ uv \in E_f\}.$$

Todėl jei v realizavo šį minimumą, briauna uv tampa tinkama. Pirmas tvirtinimas įrodytas.

Tegu priešingai antrajam teiginiui atsirado tinkama vu briauna. Turi galioti lygybė

$$h[v] = h[u] + 1.$$

Vadinasi, prieš kėlimą turėjo būti

$$h[v] > h[u] + 1.$$

Bet pagal pereito skyrelio lemą likutinių briaunų tarp viršūnių, kurių aukščiai skiriasi daugiau nei per vienetą, nėra. Be to, kėlimas nekeičia likutinių briaunų. Vadinasi, vu nepriklauso likutiniam tinklui, juo labiau ji negali būti tinkama. ◇

Algoritmo procedūrose naudojami atributai ar žymekliai. Kiekvienai viršūnei $u \in V$ sudarinėjamas neorentuotas gretimų viršūnių sąrašas $N[u]$, t.y., $v \in N[u]$, jei $uv \in E$ arba $vu \in E$. Pirmoji sąrašo viršūnė vadinama galva ir žymima $galva[N[u]]$. Po v einanti viršūnė turės atributą $sek[v]$, jis lygus NIL, jei v – paskutinė sąrašo viršūnė. Žymeklis $nagr[u]$ bus priskiriamas viršūnei iš $N[u]$, kuri tuo metu yra apdorojama.

Pagrindinė procedūra yra

Iškrauk(u):

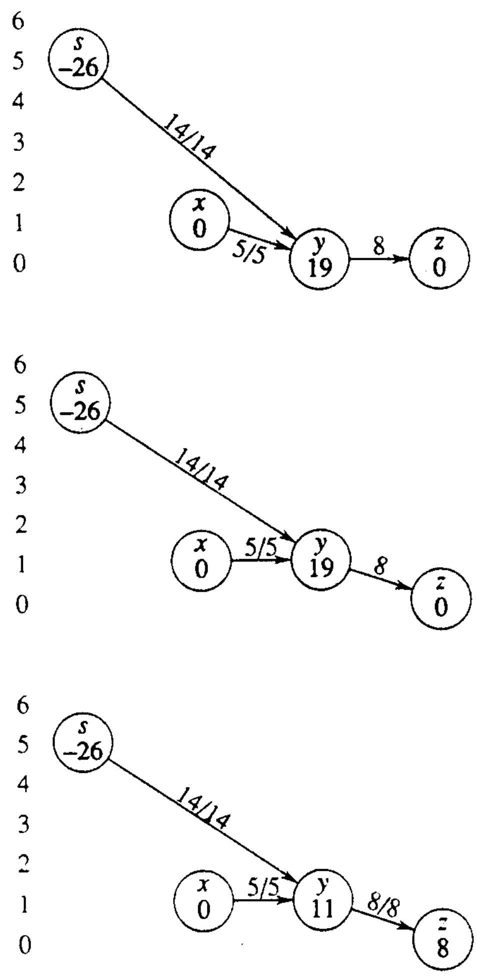
```
while e[u] > 0
    do  v ← nagr[u]
        if v = NIL
          then Kelk(u)
               nagr[u] ← galva[N[u]]
        else if c_f(uv) > 0 ir h[u] = h[v] + 1
          then Stumk(uv)
         else nagr[u] ← sek[v]
END
```

Išnagrinėkime pateiktą eskizą:

(a)

(b)

(c)



(a)

1	2	3	4
s	s	s	s
x	x	x	x
z	z	z	z

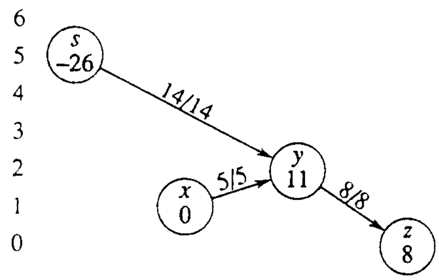
(b)

5	6	7
s	s	s
x	x	x
z	z	z

(c)

8	9
s	s
x	x
z	z

(d)

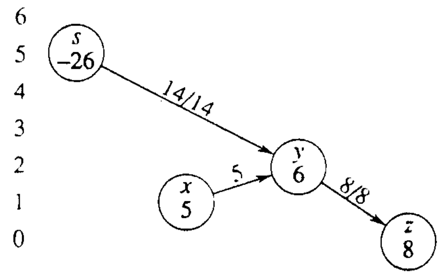


10	11
s	s
x	x
z	z

$c_f(yx) = 0-(-5) = 5$

$\exists\, yx \in E_f.$

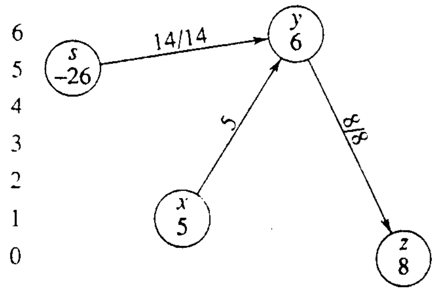
(e)



12	13	14
s	s	s
x	x	x
z	z	z

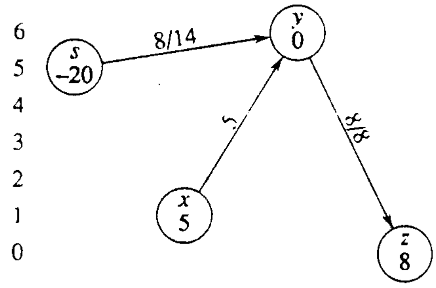
$yz \in E_f$

(f)



15
s
x
z

(g)



Svarbu, kad iškvietus **Iškrauk**(u), procedūrų **Kelk**(u) ir **Stumk**(uv) vykdymo sąlygos būtų patenkintos.

4 lema. *Jei* **Iškrauk**(u) *iškviečia* **Kelk**(u) *ir* **Stumk**(uv)*, tai tuo metu jas galima ir vykdyti.*

Įrodymas. Tvirtinimas akivaizdus dėl **Stumk**(uv), nes 1-as žingsnis garantuoja srauto perteklių, o 6-as užtikrina šios procedūros vykdymo sąlygas.

Kelk(u) vykdoma, kai $v = NIL$, vadinasi, ši viršūnė yra sąrašo gale. Kadangi žymeklis $nagr[u]$ perbėgo visą sąrašą, tai pagal 6-o žingsnio sąlygą nebuvo aptikta tinkamų briaunų, be to, procedūra **Stumk**(uv) jų nesukuria (2 lema). Pagal 3 lemą galima taikyti **Kelk** procedūrą. $\diamond$

Dabar galime nagrinėti patį algoritmą. Tariame, kad $N[u]$ – iš anksto sukurti sąrašai, o L bus sudaromas viršūnių $V \setminus \{s, t\}$ sąrašas, kurio *neiškrautų* viršūnių likutis vis trumpės, o *iškrautos* viršūnės vis bus perkeliamos į jo priekį, todėl taps jo *galva*. Viršūnių aukštis h kis, jis visą laiką bus lyginamas, todėl įveskime *senojo aukščio* kintamąjį *senauk*.

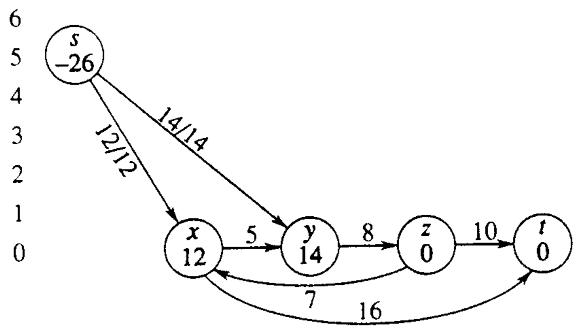
Kelk-priekin(G, s, t):

1. INIT-pr-sr(G, s)
2. $L \leftarrow V \setminus \{s, t\}$ (bet kokia tvarka)
3. **for each** $u \in V \setminus \{s, t\}$
4.   **do** $nagr[u] \leftarrow galva[N[u]]$
5. $u \leftarrow galva[L]$
6. **while** $u \neq NIL$
7.   **do** $senauk \leftarrow h[u]$
8.   *Iškrauk(u)*
9.   **if** $h[u] > senauk$
10.   **then** įrašyk u į sąrašo L piekį
11. $u \leftarrow sek[u]$

END

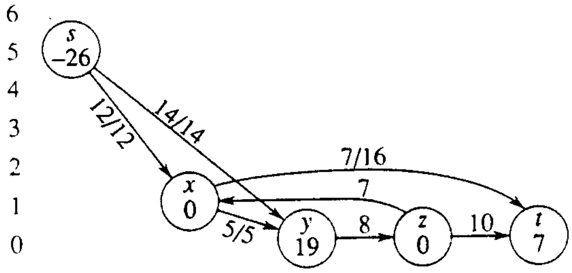
Išnagrinėkime veikimo schemą:

(a)



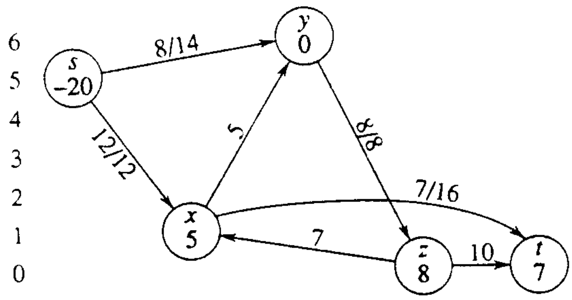
L:	x	y	z
N:	s	s	x
	y	x	y
	z	z	t
	t		

(b)



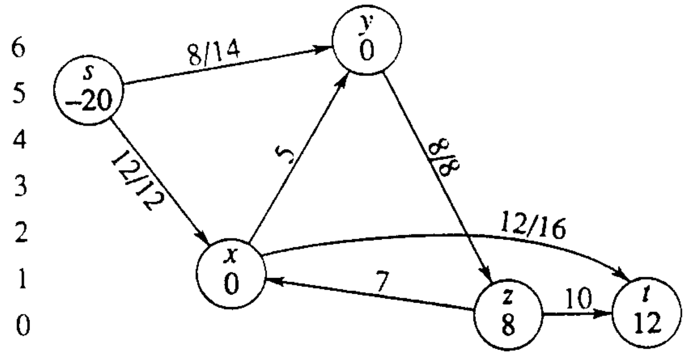
L:	x	y	z
N:	s	s	x
	y	x	y
	z	z	t
	t		

(c)



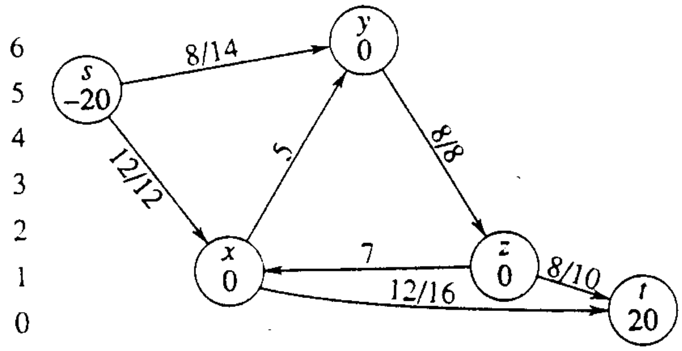
L:	y	x	z
N:	s	s	x
	x	y	y
	z	z	t
		t	

(d)



L:	y	x
N:	s	s
	x	y
	z	z
		t

(e)



L:	z	y
N:	x	s
	y	x
	t	z

Kaip matome po incializacijos sukuriamas potencialių perteklinių viršūnių sąrašas L ir 4-ame žingsnyje žymekliai $nagr[u]$ nukreipiami į gretimųjų viršūnių sąrašų galvas. Žingsnis 5 pradeda L sąrašą pirmąja viršūne. Kadangi toliau yra vykdomas perteklinio srauto iškrovimas, kurio metu gal pakisti ir viršūnės aukštis, tai senąjį aukštį pasiliekam atmintyje. Tai padaroma 7 žingsnyje. Pakelta ir iškrauta viršūnė u po to vėl atsiduria L priekyje. Toliau pereinama prie sekančios po u sąraše L viršūnės. Visa tai vykdoma tol, kol $u \neq NIL$. Kai peržiūrėtas visas sąrašas, algoritmo realizacija pasibaigia.

Norėdami įsitikinti, kad, realizavus algoritmą, yra randamas maksimalus srautas, turime įrodyti, kad mes atlikome **Bendrą-pr.sr.** stūmimo procedūrą. Ją esame aptarę anksčiau. Faktiškai pakaks įsitikinti, kad pabaigus vykdyti šį algoritmą, jokia **Stumk** ar **Kelk** procedūra nebegalima. Svarbu pastebėti, kad sąrašas L tinkamame tinkle išlieka topologiškai surūšiuotas, o srauto perteklius vis stumiamas į toliau esančias viršūnes.

5 lema. Kelk-priekin(G, s, t) *žingsniuose 6-11 atliekamos iteracijos išlaiko L sąrašo viršūnių topologinį surūšiavimą tinkamame tinkle $G_{f,h}$.*

Įrodymas. Tuoj po inicializacijos visų viršūnių, išskyrus s, aukščiai yra nuliniai. Nėra tinkamų briaunų, todėl $G_{f,h}$ yra tuščias.

Tegu jau turime topologiškai surūšiuotą L ir toliau atliekame iteracijas. Tinkamas tinklas kinta tik vykdant **Kelk** ir **Stumk** operacijas. Pagal 2 lemą pastaroji gali tik panaikinti tinkamas briaunas, tad ji neturės įtakos topologiniam viršūnių surūšiavimui.

Pagal 3 lemą pakėlus viršūnę, nebelieka įeinančių tinkamų briaunų, bet gali atsirasti išeinančių. Perkėlus tokią viršūnę į L pradžią, topologinis surūšiavimas išlieka. $\diamond$

Teorema. Kelk-priekin *algoritmas suranda maksimalųjį srautą per* $O(|V|^3)$ *laiko.*

Įrodymas. Pagal 5 lemą palaikomas topologiškai surūšiuotas sąrašas L. Jo pirmoji viršūnė yra iškraunama, ir priešsrautis nustumiamas toliau. Kai nebelieka galimybės vykdyti **Kelk** ir **Stumk** operacijų, pagal praeito skyrelio medžiagą mes žinome, kad jau yra suskaičiuotas maksimalus srautas.

Praeitame skyrelyje buvome įrodę, jog yra ne daugiau kaip $O(|V|)$ kėlimo operacijų vienai viršūnei ir $O(|V|^2)$ iš viso. Todėl yra $O(|V|^2)$ tarpų tarp dviejų kėlimų. Kiekviename tarpe **Iškrauk** kviečiama ne daugiau kaip $|V|$ kartų. Iš tiesų, jei iškvietus ją, ji jokios viršūnės nekelia (9-ame žingsnyje nurodyta sąlyga negalioja), tai mes turime iškrauti L sąrašo viršūnes. Bet jo ilgis neviršija $|V|$. Jei ji kelia viršūnę, tai sekantis **iškrauk** jau patektų į kitą tarpą. Taigi, algoritmo žingsniai, tik iškviečiantys **Iškrauk** procedūrą, o ne vykdantys pačią procedūrą, užima $O(|V|^3)$ laiko.

Vidiniai likusieji žingsniai vykdo trijų tipų veiklą. Visi kėlimai trunka $O(|V||E|)$ laiko. Čia reikia geros **Bendros-pr.st.** procedūros realizacijos. Reikia palaikyti informaciją apie likutinių tinklų viršūnių aukščius, nes yra kiek ilgesnis minimumo ieškojimo žingsnelis.

Iškrauk(u) 8-oje eilutėje yra gretimųjų viršūnių sąrašo perbėgimas. Trukmė priklauso nuo laipsnio $\delta(u)$. Tad, vienai viršūnei reikia (keliama ne daugiau $|V|$ kartų) $O(|V|\delta(u))$ laiko. Sudėję gauname $O(|V||E|)$.

Pagaliau, **Stumk** vykdoma 7-oje eilutėje. Praeitame skyrelyje matėme, kad prisotinančių yra ne daugiau kaip $O(|V||E|)$. Neprisotinantis stūmimas panaikina srauto perteklių viršūnėje. Toks stūmimas gali būti tik vienas, vieną kartą iškvietus **Iškrauk**(u). Kvietimų yra $O(|V|^3)$, vadinasi, dėl visų stūmimų sugaištama $O(|V|^3)$ laiko.

Reziumuodami gauname laiko įvertį $O(|V|^3 + |V||E|) = O(|V|^3)$.

Teorema įrodyta. $\diamond$

I DALIS. KOMBINATORINIŲ STRUKTURŲ SUSKAIČIAVIMAS

1. Adityvieji natūraliojo skaičiaus skaidiniai

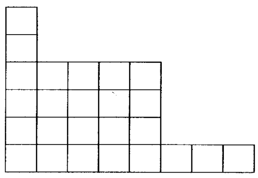
Pradėsime nuo paprastų uždavinių. Panagrinėkime klasikinį kombinatorikos ir skaičių teorijos uždavinį: *kiek kartų natūralųjį skaičių n galima užrašyti suma*

$$n = n_1 + \cdots + n_k \tag{1}$$

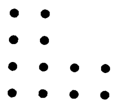
Čia k ir n_k - bet kokie natūralieji skaičiai. Dėmenų tvarką laikykime nesvarbia, todėl papildomai galime reikalauti, kad būtų $n_1 \geq \cdots \geq n_k \geq 1$. Ieškomasis skaičius vadinamas *Eulerio-Hardy-Ramanujano* vardu ir žymimas $p(n)$. Sugrupavę vienodus dėmenis, (1) lygybę galime užrašyti ir taip:

$$n = 1k_1 + \cdots + nk_n. \tag{2}$$

Dabar $k_j \geq 0$ nurodo, kiek dėmenų, lygių j buvo (1) skaidinyje. taigi, $p(n)$ - išreiškia ir vektorių $\bar{k} = (k_1, \ldots, k_n) \in \mathbf{Z}^+$, tenkinančių (2) lygybę skaičių. Skaidinius (1) arba (2) galima vaizduoti lentelėmis:



20=8+5+5+1+1



12=4+4+2+2

vadinamomis *Jungo* arba *Ferero* diagramomis.

Eulerio teorema. *Tegu* $p(0) = 1$*, tada*

$$\sum_{n\geq 0} p(n)t^n = \prod_{j\geq 1}(1 - t^j)^{-1} =: f(t), \quad t \in \mathbf{C},\ |t| < 1.$$

Euleris šią tapatybę naudojo formaliai, be griežto pagrindimo. Taip mes dažnai elgsimės ateityje, bet šį kartą pateiksime visas įrodymo detales.

Įrodymas. Tarkime m - bet koks natūralusis skaičius. Kai $|t| < 1$, galime dauginti panariui baigtinį skaičių absoliučiai konverguojančių eilučių

$$f_m(t) := \prod_{j\leq m}(1 - t^j)^{-1} = (1 + t + t^2 + \cdots) \ldots (1 + t^m + t^{2\cdot m} + \cdots) =$$

$$= \sum_{k_1,\ldots,k_m\geq 0} t^{1k_1+\cdots+mk_m} = \sum_{n\geq 0} p_m(n)t^n,$$

čia $p_m(n)$ – lygties $1k_1+\cdots+mk_m=n$ sprendinių, priklausančių $\mathbf{Z}^{+m}$ skaičius, $p_m(0)=1$. Aišku, $p_m(n)\le p(n)$ ir $p_m(n)=p(n)$, kai $1\le n\le m$. Taigi,

$$f_m(t)=1+\sum_{n=1}^{m}p(n)t^n+\sum_{n\ge m+1}p_m(n)t^n.$$

Srityje $0\le t<1$ dalinės sandaugos $f_m(t)$ konverguoja į begalinę sandaugą $f(t)$. Be to, $f_m(t)\le f(t)$, todėl

$$\sum_{n=0}^{m}p_m(n)t^n\le f_m(t)\le f(t).$$

vadinasi, eilutės

$$\sum_{n\ge 0}^{\infty}p(n)t^n,\quad \sum_{n\ge 0}^{\infty}p_m(n)t^n$$

konverguoja, pastaroji – net tolygiai m atžvilgiu. Pastebėję, jog $p_m(n)\to p(n)$, kai $m\to\infty$ su visais $n\ge 0$, srityje $0\le t<1$ pereiname prie ribos ir gauname

$$\sum_{n\ge 0}^{\infty}p(n)t^n=\lim_{m\to\infty}\sum_{n\ge 0}^{\infty}p_m(n)t^n=\lim_{m\to\infty}f_m(t)=f(t).$$

Pastebėję, kad nagrinėjamos eilutės ir sandaugos konverguoja absoliučiai srityje $|t|<1$, galime teigti, kad lygybė galioja ir šiame realiosios tiesės intervale. Pagal analizinio pratesimo principą tapatybė galioja netgi kompleksinės plokštumos vienetiniame skritulyje. $\diamond$

Pastebėkime, kad kiekviena natūraliojo laipsnio šaknis yra funkcijos $f(t)$ polius, todėl vienetinis apskritimas yra natūrali jos analizinio pratesiamumo riba. Naudojantis Koši formule galima toliau nagrinėti seką $p(n)$, bet tai – gana sudėtingi skaičiavimai. Dvidešimto amžiaus pradžioje Hardis ir Ramanudžanas, pvz., įrodė, kad

$$p(n)\sim\frac{\sqrt{3}}{4n}e^{2\pi\sqrt{n/6}},\quad n\to\infty.$$

2. Aibės skaidiniai. Belo skaičiai

Tarkime A – n elementų aibė (n aibė). Nagrinėkime visas jos išraiškas nesikertančių ir netuščių jos poaibių sąjungomis

$$A=\bigcup_{k=1}^{s}A_k,\qquad A_k\subset A,\quad A_k\cap A_j=\emptyset,\quad 1\le j<k\le s;$$

čia n – bet koks natūralusis skaičius. Tokių skaidinių kiekis vadinamas *Belo skaičiumi* ir žymimas B_n. Panagrinėkime jį.

1 teorema. *Susitarkime, kad* $B_0 = 1$. *Tada*

$$B_n = \sum_{k=1}^{n} \binom{n-1}{k-1} B_{n-k}, \quad n \geq 1.$$

Įrodymas. Galime nagrinėti aibę $A = \{1, \dots, n\}$. Bet koks A skaidinys poaibiais turi vieną poaibį Q su skaičiumi n. Tegu

$$Q = \{n\} \cup X \subset A, \quad n \notin X,$$

o X yra $(n-1)$ aibės $A \setminus \{n\}$ poaibis. Jei $|Q| = k$ – aibės Q elementų skaičius, tai kintant X galima sudaryti $\binom{n-1}{k-1}$ poaibių Q.

Pagal Belo skaičiaus apibrėžimą aibė $A \setminus Q$ gali būti išskaidoma sąjungomis B_{n-k} kartų. Kadangi visas A sąjungas galime gauti išrenkant aibes Q ir išskaidant likusį poaibį, tai

$$B_n = \sum_{k=1}^{n} \binom{n-1}{k-1} B_{n-k}.$$

$\diamond$

Kombinatorikoje dažnai naudojami *antros rūšies Stirlingo skaičiai*, $S(n,k)$, išreiškiantys n aibės skaidinių, turinčių sąjungose k poaibių, skaičių. Tad,

$$B_n = \sum_{k=1}^{n} S(n,k).$$

Stirlingo skaičius $S(n,k)$ taip pat galima apibrėžti ir lygybe

$$t^n = \sum_{k=1}^{n} S(n,k)\, t(t-1) \cdots (t-k+1).$$

Vėliau naudosime Belo skaičių eksponentinę generuojančią funkciją

$$B(t) = \sum_{n \geq 0} \frac{B_n t^n}{n!}.$$

2 teorema. $B(t) = \exp\{e^t - 1\}$.
Įrodymas. Iš 1 teoremos išplaukia

$$\begin{aligned} B'(t) &= \sum_{n \geq 1} \frac{B_n}{(n-1)!} t^{n-1} = \sum_{n \geq 1} \frac{1}{(n-1)!} \left(\sum_{k=1}^{n} \binom{n-1}{k-1} B_{n-k} \right) t^{n-1} = \\ &= \sum_{n \geq 1} \sum_{k=1}^{n} \frac{B_{n-k}}{(k-1)!(n-k)!} t^{n-k} t^{k-1} = \sum_{j \geq 1} \frac{t^j}{j!} \sum_{k \geq 0} \frac{B_k t^k}{k!} = \\ &= e^t B(t). \end{aligned}$$

Išsprendę diferencialinę lygtį

$$\frac{B'}{B} = e^t$$

su pradine sąlyga $B(0 = 1$, baigiame teoremos įrodymą. $\diamond$

Išvada (Dobinskio formulė).

$$B_n = \frac{1}{e} \sum_{k \geq 0} \frac{n^k}{k!}, \quad n \geq 1.$$

Įrodymas. Skaičiuojame $B(t)$ naudodami 2 teoremos rezultatą ir gauname

$$B(t) = e^{-1} \sum_{k \geq 0} \frac{e^{kt}}{k!} = e^{-1} \sum_{k \geq 0} \frac{1}{k!} \sum_{n \geq 0} \frac{k^n}{n!} t^n =$$
$$= e^{-1} \sum_{n \geq 0} \frac{1}{n!} \left(\sum_{k \geq 0} \frac{k^n}{k!} \right) t^n.$$

Sulyginę $B(t)$ koeficientus, gauname norimą formulę. $\diamond$

Kombinatorikoje sutinkami ir kitokie aibės skaidinių uždavinio variantai. Tarkime, reikia rasti n aibės skaidinių sąjungomis iš k poaibių, kurių pirmame yra n_1 elementų, antrame – n_2, o k-ame – n_k elementų, skaičių. Tokių skaidinių skaičius lygus polinominiam koeficientui

$$\binom{n}{n_1, \ldots, n_k} = \frac{n!}{n_1! \ldots n_k!}.$$

Čia be to, $n_1 + \cdots + n_k = n$.

3 teorema.

$$B_n = n! \sum_{\substack{\bar{k} \in \mathbf{Z}^{+n} \\ 1k_1 + \cdots + nk_n = n}} \prod_{j=1}^{n} \frac{1}{(j!)^{k_j}\, k_j!}$$

Įrodymas. Vektorius $\bar{k} = (k_1, \ldots, k_n)$ su $k_j \in \mathbf{Z}^+$ tokiai, kad $1k_1 + \cdots + nk_n = n$, nusako skaidinio struktūrą: jame yra k_j galios j poaibių. Pasidarykime k_j dėžučių, kuriose gali tilpti j, $1 \leq j \leq n$, skaičių:

$$\overbrace{(\cdot) \ldots (\cdot)}^{k_1} \ldots \overbrace{\underbrace{(\cdot, \ldots, \cdot)}_{j} \ldots \underbrace{(\cdot, \ldots, \cdot)}_{j}}^{k_j} \ldots \overbrace{\underbrace{(\cdot, \ldots, \cdot)}_{n} \ldots \underbrace{(\cdot, \ldots, \cdot)}_{n}}^{k_n}.$$

Bet kaip išdėstydami visus n skaičių į jas, t.y. panaudodami visus $n!$ kėlinių, gauname nurodytos struktūros skaidinius. Atkreipkime dėmesį į pasikartojimus. Jų priežastys yra dvi:

(i) poaibių tvarka skaidinyje yra nesvarbi;

(ii) j galios poaibio elementų tvarka irgi nesvarbi.

Kitaip tariant, naudojant įvairius kėlinius, to pačio didumo dėžutės su įrašytais skaičiais galėjo keistis vietomis ir duoti tuos pačius skaidinius. Dėl (i) priežasties kievienas skaidinys buvo pakartotas

$$k_1! \dots k_j! \dots k_n!$$

kartų, o dėl (ii) priežasties –

$$(1!)^{k_1} \dots (j!)^{k_j} \dots (n!)^{k_n}$$

kartų. Padaliję $n!$ iš šių sandaugų, gauname reikiamą formulę. $\diamond$

3. Polinomai virš baigtinio kūno

E.Galua įrodė, kad kiekvieno baigtinio kūno elementų skaičius yra pirminio skaičiaus laipsnis, kurį pažymėkime q, be to, kiekvienam q galima apibrėžti tokios eilės kūną, kurį žymėsime $\mathbf{F}_q$. Nagrinėsime polinomų virš jo žiedą $\mathbf{F}_q[x]$. Tegu $\mathbf{F}_q^*[x]$ – polinomų su vienetiniu vyriausiuoju koeficientu multiplikacinis pusgrupis. Pagal pagrindinę polinomų skaidumo teoremą kiekvienas $f \in \mathbf{F}_q^*[x]$ daugiklių tvarkos tikslumu yra išskaidomas pirminių (neskaidžių virš $\mathbf{F}_q$) polinomų sandauga

$$f = p_1 \dots p_s. \tag{1}$$

Čia $p_i \in \mathbf{F}_q^*[x]$. Tarkime, kad polinomo f laipsnis yra n ir (1) skaidinyje yra k_j pirminių polinomų, kurių laipsnis yra j, $1 \le j \le n$. Turime sąryšį

$$1k_1 + \cdots + nk_n = n. \tag{2}$$

Tegu $\pi(j)$ – j laipsnio pirminių polinomų su vienetiniu vyriausiuoju koeficientu skaičius. Rasime jo asimptotinį kitimo pobūdį, kai $j \to \infty$.

1 teorema. *Kiekvienam natūraliajam n teisinga lygybė*

$$q^n = \sum_{\substack{\bar{k} \in \mathbf{Z}^{+n} \\ 1k_1 + \cdots + nk_n = n}} \prod_{j=1}^{n} \binom{\pi(j) + k_j - 1}{k_j}.$$

Įrodymas. Pastebėkime, jog q^n lygus n laipsnio polinomų iš $\mathbf{F}_q^*[x]$ skaičiui. Visus tokius polinomus suskirstykime į klases. Tegu vieną klasę sudaro n laipsnio polinomai, kurių (1) skaidinyje yra k_j j-o laipsnio pirminių polinomų. Vektorius $\bar{k}$ vadinamas šios klasės polinomų *struktūros* vektoriumi. Jis tenkins (2) sąlygą ir vienareikšmiškai apibrėš nagrinėjamą klasę.

Kadangi kiekvieną polinomą galima suvokti kaip pirminių polinomų (1) sandaugą, suskaičiuokime, kiek tokių sandaugų galima sudaryti. Pirminius polinomus, kurių laipsnis yra j, galime imti su pakartojimais iš visos jų aibės, turinčios $\pi(j)$ elementų. Pagal kartotinių derinių apibrėžimą gauname

$$\binom{\pi(j)+k_j-1}{k_j} = (-1)^{k_j}\binom{-\pi(j)}{k_j} \tag{3}$$

galimybių. Skirtingų laipsnių polinomų rinkimas atliekamas nepriklausomai, todėl nagrinėjamos klasės polinomų skaičius lygus šių koeficientų sandaugai, o visų n laipsnio plolinomų skaičių gausime sudėję šias sandaugas pagal struktūros vektorius $\bar{k}$, tenkinančius (2) sąlygą. ◇

2 teorema. *Jei* $z \in \mathbf{C}$, $\quad |z| < q^{-1}$, *tai*

$$\sum_{n\geq 0} q^n z^n = (1-qz)^{-1} = \prod_{j\geq 1}(1-z^j)^{-\pi(j)}.$$

Įrodymas. Pasinaudokime apibendrintąja Niutono binomo ir (3) formulėmis. Gauname

$$(1-z^j)^{-\pi(j)} = \sum_{k\geq 0}\binom{\pi(j)+k-1}{k} z^{jk}.$$

Formaliai dauginkime šias eilutes pagal $j \geq 1$ ir gautuosius dėmenis grupuokime pagal vienodus z laipsnius. Gauname

$$\prod_{j\geq 1}(1-z^j)^{-\pi(j)} = \sum_{k_1,k_2,\dots\geq 0}\binom{\pi(1)+k_1-1}{k_1} z^{1k_1}\binom{\pi(2)+k_2-1}{k_2} z^{2k_2}\cdots =$$
$$= \sum_{n\geq 0} z^n \Bigg(\sum_{\substack{\bar{k}\in\mathbf{Z}^{+n}\\ 1k_1+\cdots+nk_n=n}} \prod_{j=1}^{n}\binom{\pi(j)+k_j-1}{k_j}\Bigg).$$

Pagal 1 lemą apskliaustasis koeficientas lygus q^n. Taigi formali lygybė yra įrodyta. Panašių formalių operacijų pagrindimą, esame aptarę skyrelyje apie natūraliojo skaičiaus skaidinius. ◇

Skaičių teorijoje yra apibrėžiama Miobiuso funkcija:

$$\mu(m) = \begin{cases} 1, & \text{jei } m=1; \\ 0, & \text{jei egzistuoja pirminio skaičiaus kvadratas, dalijantis } m; \\ (-1)^k, & \text{jei } m \text{ išsiskaido } k \text{ skirtingų pirminių skaičių sandauga.} \end{cases}$$

Viena iš jos savybių yra išreiškiama apgrežimo formulėje.

Lema (Miobiuso apgrežimo formulė). *Lygybės*

$$a_n = \sum_{m|n} b_n \quad \textit{ir} \quad b_n = \sum_{m|n}\mu(m)a_{n/m}, \qquad a_n, b_m \in \mathbf{C},$$

yra ekvivalenčios. Čia sumuojama pagal visus natūraliuosius n daliklius.

Įrodymas. ◇

3 teorema. *Su visais natūraliiaisiais skaičiais n teisinga formulė*

$$\pi(n) = \frac{1}{n}\sum_{m|n}\mu(m)q^{n/m} = q^n/n + O(nq^{n/2}).$$

Įrodymas. Logaritmuodami 2 teoremoje gautą generuojančios funkcijos ičraišką, gauname

$$\sum_{n\geq 1}\frac{(qn)^n}{n} = \sum_{j\geq 1}\sum_{k\geq 1}\frac{t^{kj}}{k} = \sum_{n\geq 1}\Big(\sum_{j|n} j\pi(j)\Big)\frac{t^n}{n}.$$

Vadinasi,

$$q^n = \sum_{j|n} j\pi(j).$$

Taigi, pirmasis teoremos teiginys išplaukia iš Miobiuso apgrežimo formulės. Įvertis gaunamas atskyrus dėmenį, kai $m = 1$, ir trivialiai įvertinus kitus dėmenis. ◇

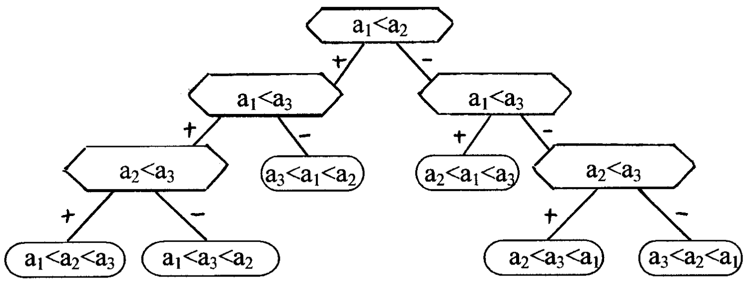
Užduotis. Įrodykite, kad pusgrupyje $\mathbf{F}_q^*[x]$ n-ojo laipsnio polinomų, neturinčių kartotinių pirminių daugiklių (*bekvadračių*) skaičius

$$\tilde{p}(n) = \sum_{\substack{\bar{k}\in\mathbf{Z}^{+n}\\ 1k_1+\cdots+nk_n=n}}\ \prod_{j=1}^{n}\binom{\pi(j)}{k_j}.$$

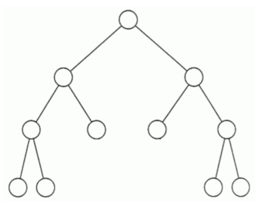
4. Rūšiavimo algoritmas ir binarieji medžiai

Informatika irgi ikelia kombinatorinių uždavinių. Ypač aktualios yra algoritmų teorijos problemos. Nagrinėkime bene populiariausią duomenų rūšiavimo uždavinį.

Tarkime, kad reikia sutvarkyti n skirtingų duomenų sąrašą $\{a_1, \ldots, a_n\}$ pagal kokį nors požymio (rakto) didėjimo eilę. Paprastumo dėlei laikykime, kad duomenys yra realieji skaičiai, todėl raktas yra jų didumas. Kai $n = 3$, algoritmas galėtų būti toks, kaip pavaizduota paveiksle.



Lakoniškai iliustruodami, gauname plokščiąjį grafą, vadinamą *binariuoju medžiu*:



Tokius medžius galime apibrėžti rekursyviai, t. y. naudojant pilnosios matematinės indukcijos principą.

Apibrėžimas. Medis T yra binarusis, jeigu

(i) jame yra viršūnė (vadinama *šaknimi*), sudaranti T, arba

(ii) ji yra sujungta su *kairiuoju* ir *dešiniuoju* binariaisias medžiais.

Aišku, (i) atveju turime tuščiąjį pirmos eilės medį, sudarytą iš vienos viršūnės, o (ii) atveju kairysis ir dešinysis medžiai yra mažesnių eilių negu T, todėl jų apibrėžimas galėjo būti laikomas indukcine prielaida. Žinoma, binarųjį medį galima nusakyti išvardijant jo būdingus bruožus. Kai jo eilė yra didesnė už vienetą, reikalaujama, kad jis turėtų savybes:

(iii) jame yra viena antrojo laipsnio viršūnė, laikoma *šaknimi*;

(iv) kitų viršūnių laipsniai yra lygūs 1 (jos vadinamos *lapais*) arba 3 (*vidinės viršūnės*);

(v) briaunų išvedimas kairėn ar dešinėn yra įskaitomas, t.y. jos laikomos skirtingomis.

Atkreipkime dėmesį į tai, kad takas nuo šaknies iki lapo (medyje toks takas yra vienintelis) binariajame medyje, vaizduojančiame algoritmą, atitinka kažkokio sąrašo duomenų surūšiavimą. Rūšiuojant n duomenų, kad programa veiktų visais įmanomais atvejais, iš viso lapų turi būti ne mažiau negu $N := n!$ lapų. Kiekvieną binarų medį su šia savybe atitinka rūšiavimo algoritmas, todėl svarbu sužinoti binariųjų medžių, turinčių N lapų, skaičių C_N. Susitarkime, kad $C_1 = 1$. Skaičiai C_N vadinami *Katalano* vardu. Rasime jų savybių.

Teorema. *Katalano skaičiai C_N tenkina rekurentųjį sąryšį*

$$C_N = \sum_{k=1}^{N-1} C_k C_{N-k}, \quad C_1 = 1. \tag{1}$$

Be to,

$$C_N = \frac{1}{N}\binom{2N-2}{N-1} \sim \frac{2^{2(N-1)}}{N^2\sqrt{\pi}}, \tag{2}$$

kai $N \to \infty$.

Įrodymas. Tegu $N > 1$. Iš binaraus grafo atėmę jo šaknį, gauname du binariuosius medžius. Tarkime, kairiajame iš jų yra k lapų, o dešiniajame – $(N-k)$ lapų. Čia k gali būti bet kuris iš skaičių $1, \dots, N-1$. Pagal Katalano skaičiaus apibrėžimą galime nepriklausomai sudaryti C_k kairiųjų medžių ir C_{N-k-1} dešiniųjų. Sudėję pagal k, baigiame (1) lygybės įrodymą. Kitos lygybės įrodymui panaudojame generuojančią sekos C_N funkciją

$$F(t) := \sum_{N\geq 1} C_N t^N.$$

Kadangi

$$F(t)^2 = \sum_{N\geq 2}\left(\sum_{k=1}^{N-1} C_k C_{N-k}\right) t^N,$$

todėl

$$F(t) = t + F(t)^2$$

ir $F(0) = 0$. Vadinasi,

$$F(t) = \frac{1}{2}\left(1 - (1-4t)^{1/2}\right).$$

Naudodami apibendrintąją Niutono binomo formulę, randame koeficientą prie t^N. Jis lygus

$$C_N = -\frac{1}{2}\binom{1/2}{N}(-4)^N = \frac{1}{2}\frac{(2N-3)!!\,4^N}{2^N N!} = \frac{(2N-2)!}{(N-1)!N!}.$$

Taigi, (2) lygybė įrodyta. Teoremoje pateikta asimptotika išplaukia iš Stirlingo formulės:

$$\sqrt{2\pi n}(n/e)^n \leq n! \leq e^{1/12n}\sqrt{2\pi n}(n/e)^n.$$

◇

5. Vidutinis kelio ilgis greitojo rūšiavimo algoritme

Algoritmo kokybė priklauso nuo vidutinio panaudotų duomenų palyginimų skaičiaus, laikant, jog duomenų sąraše dydžių tvarka yra vienodai galima. Kaip ištirti šį algoritmo parametrą, ypač kai duomenų skaičius didėja? Binariajame medyje jį atitinka vidutinis tako nuo šaknies iki lapo ilgis. Kaip ir anksčiau rūšiuokime n skirtingų duomenų sąrašą $\{a_1, \dots, a_n\}$ pagal rakto didėjimą. Tegu duomenų raktas yra jų didumas. Panagrinėkime vieną iš populiariausių rūšiavimo algoritmų, vadinamą *greituoju* arba *Hoare's* algoritmu:

(i) žingsnis: imame pirmąjį sąrašo elementą $a = a_1$ ir sudarome du dalinius duomenų sąrašus L^- ir L^+ tokius, kad L^- duomenys būtų mažesni už a, o L^+ duomenys būtų didesni;
(ii) surūšiuojame L^- ir L^+;
(iii) turėdami surūšiuotus L^- ir L^+ bei jų tarpe a, baigiame procedūrą.

Teorema. *Tarkime, kad q_n - n duomenų sąrašo elementų vidutinis palyginimų skaičius naudojant Hoarės algoritmą. Tada*

$$q_n = 2n \log n + O(n), \quad n \geq 2.$$

Įrodymas. Pastebėkime, kad pirmajame algoritmo žingsnyje visada atliekame $n-1$ duomenų palyginimų. Antrame žingsnyje atskirta aibė L^- su vienoda tikimybe gali turėti $k = 1, \dots, n-1$ duomenų, o L^+ - $(n-1-k)$ duomenų. Jas sutvarkydami vidutiniškai naudojame $q_k + q_{n-1-k}$ palyginimų. Vidurkinant pagal k ir prisiminę pirmąjį žingsnį, gauname

$$q_n = n - 1 + \frac{1}{n}\sum_{k=1}^{n-1}(q_k + q_{n-1+k}) = n - 1 + \frac{2}{n}\sum_{k=1}^{n-1} q_k. \tag{2.1}$$

Aišku, kad $q_1 = 0$. Naudodami generuojančias funkcijas randame sekos q_n asimptotiką. Pažymėkime

$$Q(t) = \sum_{n=1}^{\infty} q_n t^n.$$

Padauginę panariui (2.1) iš nt^n ir sudėję pagal $n \geq 1$, gauname

$$\sum_{n\geq 1} n q_n t^n = \sum_{n \geq 1} n(n-1)t^n + 2\sum_{n\geq 1}\left(\sum_{k=1}^{n-1} q_k\right)t^n. \tag{2.2}$$

Matome, jog kairėje pusėje esanti eilutė lygi $tQ'(t)$. Ieškodami pirmosios eilutės dešinėje pusėje išraiškos, porą kartų panariui diferencijuojame begalinę geometrinę progresiją

$$\sum_{n\geq 0} t^n = (1-t)^{-1}, \quad |t| < 1.$$

Gauname, kad ieškoma eilutė lygi $2t^2/(1-t)^3$. Paskutinės eilutės (2.2) formulėje ieškome naudodami lygybę

$$\sum_{n\geq 1} t^n \sum_{n\geq 1} q_n t^n = \sum_{n\geq 1} \Big(\sum_{k=1}^{n-1} q_k\Big) t^n.$$

Taigi, š (2.2) išplaukia

$$tQ'(t) = \frac{2t^2}{(1-t)^3} + \frac{2tQ(t)}{1-t} \tag{2.3}$$

Išsprendę šią pirmos eilės diferencialinę lygtį, kai patenkinama pradinė sąlyga $Q(0) = 0$, gauname

$$\begin{aligned} Q(t) &= -2\frac{t + \log(1-t)}{(1-t)^2} = \\ &= 2(1 + 2t + 3t^2 + \cdots)(\frac{t^2}{2} + \frac{t^3}{3} + \cdots). \end{aligned}$$

Iš čia gauname

$$q_n = 2\sum_{k=2}^{n} \frac{1}{k}(n-k+1) = 2n\log n + O(n), \quad n \geq 2.$$

Teorema įrodyta. $\diamond$

Žinoma, kad bet kokiame rūšiavimo algoritme vidutiniškai reikia ne mažiau negu $\log_2 N! \approx 1,44...n\log n$ sąrašo elementų palyginimų.

6. Medžių skaičius. Cayley'io teorema.

Panagrinėkime kitokius negu binarieji medžiai grafus. Tarkime $G = (V, E)$ – grafas, kurio viršūnių aibė V yra netuščia, o briaunų aibė E yra sudaryta iš nesutvarkytųjų porų $e = xy$, čia $x, y \in V,\ x \neq y$. Grafai $G = (V, E)$ ir $G' = (V', E')$ vadinami *izomorfiškais*, jei egzistuoja bijekcija $\phi : V \to V'$ tokia, kad $xy \in E$ tada ir tik tada, kada $\phi(x)\phi(y) \in E'$. Skaičiuojant fiksuotos eilės $|V| = n$ grafų skaičių izomorfiškus grafus laikome lygiais. Įdomesnis yra grafų su sunumeruota viršūnių aibe, vadinamų **numeruotaisiais** grafais, atvejis. Dabar izomorfizmas turi išlaikyti ir numeraciją, t.y., jei x yra i-toji G grafo viršūnė, tai izomorfiškame G' grafe $\phi(x)$ turi būti irgi i-tąja viršūne. Pradėkime nuo paprasto teiginio.

1 teorema. *Iš viso yra*

$$2^{n(n-1)/2}$$

neizomorfiškų numeruotųjų n eilės grafų.

Įrodymas. Pastebėkime, kad uždavinys ekvivalentus pilnojo numeruoto grafo K^n pografių skaičiaus nustatymui. Bet kurios briaunos $e = x_i x_j$ galų numeriai nurodo, kurias viršūnes ji jungia, todėl skaičiuojamus pografius vienareikšmiškai apibrėžia galimi briaunų poaibiai. Pilnajame grafe yra $n(n-1)/2$ briaunų, todėl briaunų aibės poaibių skaičius lygus teoremoje nurodytam dydžiui. $\diamond$

Ketvirtame skyrelyje radome tam tikrų neizomorfiškų nemumeruotų plokščių medžių skaičių. Kaip elgtis numeruotų medžių atveju? 1889 metais Cayley apskaičiavo neizomorfiškų numeruotų n tos eilės medžių kiekį $T(n)$? Pradžioje įsitinkiname, jog yra

$$\frac{4!}{2} + 4 = 16$$

skirtingų 4-os eilės medžių. Savarankiškai panagrinėkite didesnės eilės medžius.

Cayley'io teorema. *Iš viso galime sudaryti n^{n-2} neizomorfiškų numeruotų n eilės medžių.*

1 -sis įrodymas (Prüfer'io). Tarkime $\mathcal{G}$ - nagrinėjamų medžių aibė. Kadangi sekų aibės

$$\{(a_1, \dots, a_{n-2}) : \ 1 \leq a_i \leq n, \, 1 \leq i \leq n-2\} =: \mathcal{A}$$

galia yra n^{n-2}, pakaks rasti bijektyvų atvaizdį $\mathcal{A} \to \mathcal{G}$.

Kai $n \leq 2$, teiginys akivaizdus.

Tegu toliau $n > 2$. Medžiui $G = (V, E)$, kurios viršūnių aibė sunumeruota, $V = \{x_1, \dots, x_n\}$, vienareikšmiškai priskirsime seką $\alpha = (a_1, \dots, a_{n-2}) \in \mathcal{A}$, vadinamą medžio **Prüfer'io kodu**. Pradėkime nuo medžio galinės viršūnės, kurios laipsnis lygus 1. Tokios viršūnės egzistuoja, nes kiekviena briauna turi dvi viršūnes ir todėl

$$\sum_{i=1}^{n} \delta(x_i) = 2(n-1).$$

Iš kelių tokių viršūnių išrinkime tą, kurios indeksas yra mažiausias. Tegu tai viršūnė x_{b_1}, o a_1 - indeksas viršūnės, gretimos pirmajai. Grafas $G - x_{b_1}$ yra $n-1$ eilės medis, todėl procesą galima kartoti, kol viršūnių, likusių grafe, skaičius yra didesnis už 2. Kai šis skaičius lygus 2, mes jau esame sudarę vienintelę seką $(a_1, \dots, a_{n-2})$.

Atvirkščiai, ar bet kokiai sekai $\alpha = (a_1, \dots, a_{n-2}) \in \mathcal{A}$ galima vienareikšmiškai priskirti medį? Atidėkime n viršūnių ir brėžkime norimą medį, vadovaudamiesi žemiau nurodytomis taisyklėmis:

a) jei b_1 - mažiausias iš bent dviejų natūraliųjų skaičių (iš $1, ..., n$), nepasirodžiusių sekoje α, tada junkime x_{b_1} su x_{a_1};

b) aibę $\{1, \dots, n\}$ pakeiskime $\{1, \dots, n\} \setminus \{b_1\}$, o α - seka $(a_2, \dots, a_{n-2})$;

c) procesą kartojame, kol išsemiame visą seką (tuo pačiu nubrėžiame $n-2$ grafo briaunas);

d) tarpusavyje sujungiame dvi likusias viršūnes.

Taip vienareikšmiškai gautasis grafas yra medis, nes jis jungia visas n viršūnių, o jo didumas yra $n-1$.

Kadangi abu nagrinėti atvaizdžiai yra vienas kito atžvilgiu yra atvirkštiniai, teorema įrodyta.

Grafų teorijai artimesnis kitas Cayley'io teoremos įrodymo būdas.

Antrasis teoremos įrodymas. Tarkime $T(n,k)$ - kiekis n tos eilės medžių, kuriuose fiksuota viršūnė $x \in V$ yra k-ojo laipsnio, $2 \le k \le n-1$. Viršūnės numeris nesvarbus, jo neminėsime. Išvesime sąryšį tarp $T(n,k)$ ir $T(n,k-1)$.

Imame medį G, kuriame $d(x) = k-1$. Jame išmeskime briauną uv, neincidenčią su x. Grafas skilo į du pomedžius, viename iš jų yra viršūnės x ir u arba x ir v. Tarkime, yra pirmasis atvejis. Sujungę dabar x su v, gauname vėl medį G', kuriame $d(x) = k$. Porą (G, G') pavadinkime *junginiu* ir suskaičiuokime jų kiekį dviem būdais. Kadangi grafui G mes galime sudaryti tiek G', kiek yra briaunų su aukščiau minėtomis savybėmis, tai vienam G mes turime $n-1-(k-1) = n-k$ partnerių. Taigi, iš viso yra $(n-k)T(n,k-1)$ junginių.

Skaičiuokime tą patį skaičių kitu būdu, pradėdami nuo G', kuriame $d(x) = k,\ k \ge 2$. Tarkime $x_1, \dots, x_k$ - gretimos x viršūnės. Paeiliui išmesdami briaunas $xx_i,\ i = 1, \dots k$, mes "atskeltume" pomedžius $T_1, \dots, T_k$, kurių eilės tegu bus $n_1, \dots, n_k$,

$$n_1 + \cdots + n_k = n - 1. \tag{1}$$

Grafo G' partnerį junginyje dabar konstruojame tokiu būdu:

a) išmetame xx_1, o vėliau viršūnę x_1 sujungiame su bet kokia iš viršūnių, nepriklausančių T_1 (turime $n-1-n_1$ galimybių);

b) tą patį kartojame su $T_2, \dots, T_k$.

Atsižvelgę į grafų G' kiekį $T(n,k)$ ir (1) iš viso gauname junginių

$$\sum_{i=1}^{n} T(n,k)(n-1-n_i) = (n-1)(k-1)T(n,k).$$

Sulyginę abi junginių skaičiaus formules, gauname

$$(n-1)(k-1)T(n,k) = (n-k)T(n,k-1).$$

Kai $k = 1$, ši rekurenčioji formulė irgi teisinga. Jos nagrinėjimui galime panaudoti akivaizdų faktą, kad $T(n, n-1) = 1$ (**žvaigždinio** grafo atvejis). Gauname

$$T(n,k) = \binom{n-2}{k-1}(n-1)^{n-k-1}.$$

Sudėdami šias lygybes, išvedame medžių kiekio $T(n)$ formulę

$$T(n) = \sum_{k=1}^{n-1} T(n,k) = \sum_{k=1}^{n-1} \binom{n-2}{k-1}(n-1)^{n-k-1} = ((n-1)+1)^{n-2} = n^{n-2}.$$

Teorema įrodyta. ◇

Numeruotą medį su viena išskirta viršūne, *šaknimi*, vadinsime *šakniniu* medžiu.

Išvada. *Yra* $d_n := n^{n-1}$ *šakninių* n *eilės medžių.*

Įrodymas. Kiekvieno medžio, kurių kiekį nusako Cayley'io teorema, šaknimi gali būti bet kuri viršūnė. $\diamond$

Šakniniai numeruoti medžiai vadinami *Cayley'io* vardu. Baigtinis jų rinkinys vadinamas *šakniniu mišku.* Jį sudarančių medžių šaknų rinkinys laikomas *miško šaknimi.* Kai miško medžių tvarka yra įskaitoma, miškas vadinamas *plokščiuoju.* Jo šaknis bus sutvarkytasis medžių šaknų rinkinys.

2 teorema. *Jei* q_n *–* n *eilės šakninių miškų skaičius, tai*

$$q_n = (n+1)^{n-1}. \tag{2}$$

Įrodymas. Imkime $(n+1)$-os eilės Cayley'io medį ir atimkime jo šaknį, turėjusią numerį $j \in \{1, \ldots, n+1\}$. Medis skyla į n eilės mišką. Sunumeruokime jo viršūnes pirmaisiais n natūraliųjų skaičių. Tuo tikslu buvusius viršūnių indeksus, didesnius už j, sumažinkime vienetu. Nepriklausomai nuo buvusio j, gauname vieną numeruotą n eilės mišką. Taigi, iš $(n+1)$-o $(n+1)$-os eilės medžio gavome vieną mažesnės eilės mišką.

Atvirkščiai, turėdami n eilės šakninį mišką iš keleto Cayley'io medžių, įvedame papildomą viršūnę, ir ją briaunomis sujungiame su medžių šaknimis. Priskirdami paeiliui papildomajai viršūnei numerius $j = 1, \ldots, n+1$ ir buvusių viršūnių indeksus, ne mažesnius negu j padidindami vienetu, gautume $(n+1)$-ą $(n+1)$-os eilės Cayley'io medį. Vadinasi, $q_n = d_{n+1}/(n+1)$. Dabar (2) išplaukia iš Cayley'io teoremos. $\diamond$

7. Simetrinė grupė

Viena iš svarbiausių kombinatorinių struktūrų yra simetrinė grupė. Paminėsime porą jos savybių. Bijektyvus n aibės X atvaizdis σ į ją pačią vadinamas *keitiniu.* Paprastumo dėlei, tegu $X = \{1, \ldots, n\}$, tada σ patogu žymėti lentele

$$\sigma = \begin{pmatrix} 1 & 2 & \ldots & n \\ 1\sigma & 2\sigma & \ldots. & n\sigma \end{pmatrix},$$

kurioje $j\sigma$ yra elemento j vaizdas, $j = 1, \ldots, n$. Tegu $\mathbf{S}_n$ visų keitinių aibė, o σ_1, σ_2 du jos atstovai. Lygybė

$$j(\sigma_1\sigma_2) = (j\sigma_1)\sigma_2, \quad 1 \le j \le n$$

apibrėžia algebrinę operaciją, vadinamą *keitinių daugyba.* Algebroje įrodoma, kad $\mathbf{S}_n$ šios operacijos atžvilgiu yra grupė, vadinama *simetrine grupe.* Jos eilė yra $n!$.

Jei keitinys s skirtingų skaičių $j_1, \ldots, j_s$, $1 \le s \le n$, vaizduoja cikliškai, t.y.

$$j_1 \mapsto j_2 \mapsto \ldots j_s \mapsto j_1,$$

o kiti skaičiai paliekami vietoje, tai jį vadiname s ilgio *ciklu.* Tokį keitinį patogu žymėti viena *slankiųjų simbolių* eilute

$$(j_1\, j_2\, \ldots\, j_s). \tag{1}$$

1 teorema. *Yra $(s-1)!$ s ilgio ciklų.*

Įrodymas. Anksčiau pastebėjome, kad galime sudaryti $s!$ kėlinių iš s elementų, kurie pagal (1) susitarimą duotų ciklus. Dabar atkreipkime dėmesį į tai, kad žymėdami tą patį ciklą, galėjome pradėti nuo bet kurio elemento ir cikliškai tęsti toliau. Vadinasi, darant iš visų kėlinių ciklus s kartų pasikartos tas pats ciklas. $\diamond$

Jei $\{j_1, \ldots, j_s\} \cap \{i_1, \ldots, i_m\} = \emptyset$, tai ciklai

$$(j_1, \ldots, j_s), \quad (i_1, \ldots, i_m)$$

vadinami *nepriklausomais.*

2 teorema. *Kiekvieną keitinį galima išreikšti nepriklausomų ciklų sandauga*

$$\sigma = \kappa_1 \cdots \kappa_w. \tag{2}$$

Čia $w = w(\sigma)$ – ciklų skaičius. Be to, tokia išraiška yra vienintelė daugiklių užrašymo tvarkos tikslumu.

Įrodymas. Panaudokime algoritmizuotus samprotavimus. Jeigu j dar nėra priskirtas nagrinėjamo keitinio σ ciklui, tai raskime mažiausią natūralųjį skaičių m su savybe $j\sigma^m = j$. Toks $m \leq n$ egzistuoja, nes turime ne daugiau negu n skirtingų skaičių sekoje $j\sigma^k$, $k \geq 1$. Sudarome ciklą

$$\kappa := (j\ j\sigma\ \ldots\ j\sigma^{m-1}).$$

Tai iš tiesų yra ciklas, nes lygybė $j\sigma^k = j\sigma^l$ su $1 \leq k < l \leq m = 1$, dėka $j = j\sigma^{l-k}$, prieštarautų m minimalumui. Skaičius m būtų šio ciklo ilgis.

Toliau taip pat skaičius $X \backslash \{j, j\sigma, \ldots, j\sigma^{m-1}\}$ skirstytume į ciklus. Naujasis ciklas būtų nepriklausomas nuo prieš tai sudaryto. Iš tiesų, jei i, $i^p = i$, – skaičius, nepriklausantis pirmajam ciklui, tai lygybė

$$j\sigma^k = i\sigma^l \tag{3}$$

vestų prie sąryšio $j\sigma^{k+p-l} = i$, rodančio, kad i turėtų priklausyti pirmajam ciklui. Prieštara akivaizdi. Išsėmę visus X skaičius, baigiame skaidinio egzistavimo įrodymą.

Jei egzistuotų pora skirtingų skaidinių ciklais, besiskiriančių ne tik išdėstymo tvarka, tai turėtų egzistuoti bent vienas elementas, patenkantis į du ciklus ir todėl jam rastume dvi išraiškas, tarkim, (3). Tai vėl duoda prieštarą. $\diamond$

Kombinatorikai svarbu, kokio ilgio ir kiek ciklų sudaro keitinį. Pažymėkime k_j j ilgio ciklų (2) skaidinyje, $1 \leq j \leq n$. Aišku,

$$1k_1 + \cdots + nk_n = n \tag{4}$$

o ciklų kiekis lygus $w(\sigma) = k_1 + \cdots + k_n$. Vektorių $\bar{k} := \bar{k}(\sigma) = (k_1, \ldots, k_n) \in \mathbf{Z}^{+n}$ vadinsime keitinio σ *struktūros vektoriumi.* Jis turi ir algebrinę prasmę.

Du simetrinės grupės $\mathbf{S}_n$ keitiniai σ ir σ_1 vadinami *jungtiniais*, jeigu egzistuoja $\tau \in \mathbf{S}_n$ toks, kad

$$\sigma = \tau\sigma_1\tau^{-1}. \tag{5}$$

Čia τ^{-1} keitiniui τ atvirkštinis keitinys.

3 teorema. *Du keitiniai yra jungtiniai tada ir tik tada, jei jų struktūros vektoriai sutampa.*

Įrodymas. Tarkime $\kappa = (x_1, \ldots, x_k)$ - keitinio σ_1 ciklas, σ_1 yra jungtinis su σ ir galioja (5) sąryšis. Turime

$$x_1 = x_k\sigma_1,\ x_2 = x_1\sigma_1,\ \ldots,\ x_k = x_{k-1}\sigma_1.$$

Pažymėkime $y_j = x_j\tau^{-1}$, $1 \leq j \leq k$. Patikriname, jog $(y_1, \ldots, y_k)$ - keitinio σ ciklas:

$$y_j\sigma = y_j(\tau\sigma_1\tau^{-1}) = (y_j\tau)(\sigma_1\tau^{-1}) = (x_j\sigma_1)\tau^{-1} = x_{j+1}\tau^{-1} = y_{j+1},$$

jei $j = 1, \ldots, k-1$, ir

$$y_k\sigma = (x_k\sigma_1)\tau^{-1} = x_1\tau^{-1} = y_1.$$

Išreiškę iš (5) keitinį σ_1 per σ, panašiai įsitikintume, jog ir bet koks σ ciklas atitinka σ_1 to paties ilgio ciklą.

Tarkime dabar, kad $\bar{k}(\sigma) = \bar{k}(\sigma_1)$. Imkime jų išraiškas ciklais. Tegu $(x_1x_2\ldots x_s)$ - bendrasis ciklas keitinyje σ, o $(y_1y_2\ldots y_s)$ - keitinyje σ_1. Atitinkamai sutvarkius ciklų išdėstymo eilę, sudarykime keitinį

$$\tau = \begin{pmatrix} \ldots x_1\, x_2\, \ldots\, x_s \ldots \\ \ldots y_1\, y_2\, \ldots\, y_s \ldots \end{pmatrix}.$$

Patikrinkime (5) formulę. Du aibės atvaizdžiai lygūs, jei jų reikšmės tuose pačiuose taškuose sutampa. Kaip vaizduoja x_j atvaizdis σ žinom, o į ką atvaizduoja tuos pačius skaičius $\tau\sigma_1\tau^{-1}$, surandame:

$$x_j(\tau\sigma_1\tau^{-1}) = (x_j\tau)\sigma_1\tau^{-1} = (y_j\sigma_1)\tau^{-1} = y_{j+1}\tau^{-1} = x_{j+1},$$

jei $j = 1, \ldots, s-1$. Panašiai gautume $x_k(\tau\sigma_1\tau^{-1}) = x_1$. Rastosios reikšmės sutampa su x_j vaizdais, naudojant σ. (5) lygybė įrodyta. $\diamond$

Jungtinmiai elementai grupėje $\mathbf{S}_n$ sudaro atskirą klasę, ją vienareikšmiškai atitinka struktūros vektorius $\bar{k}$, kurio sveikos neneigiamos koordinatės tenkina (4) lygybę.

4 teorema. *Jei simetrinės grupės $\mathbf{S}_n$ jungtinių elementų elementų klasė $S(\bar{k})$ apibrėžiama struktūros vektoriumi $\bar{k}$, tai joje yra*

$$|S(\bar{k})| = n! \prod_{j=1}^{n} \frac{1}{k_j! j^{k_j}}$$

keitinių.

Įrodymas. Pasinaudojame 2.3 teoremos įrodymo idėja. Turėdami struktūros vektorių, pasidarykime k_j dėžučių, kuriose gali tilpti j, $1 \leq j \leq n$, skaičių:

$$\overbrace{(\cdot)\ldots(\cdot)}^{k_1}\ldots\overbrace{\underbrace{(\cdot,\ldots,\cdot)}_{j}\ldots\underbrace{(\cdot,\ldots,\cdot)}_{j}}^{k_j}\ldots\overbrace{\underbrace{(\cdot,\ldots,\cdot)}_{n}\ldots\underbrace{(\cdot,\ldots,\cdot)}_{n}}^{k_n}.$$

Bet kaip išdėstydami visus n skaičių į jas, t.y. panaudodami visus $n!$ kėlinių, gauname nurodytos struktūros keitinius. Atkreipkime dėmesį į pasikartojimus. Jų priežastys yra dvi:

(i) ciklų tvarka keitinyje yra nesvarbi;

(ii) j ilgio ciklą galima užrašyti j būdų, keičiant cikliškai jo elementus (žr. 1 teoremos įrodymą).

Kitaip tariant, naudojant įvairius kėlinius, to pačio didumo dėžutės su įrašytais skaičiais galėjo keistis vietomis ir duoti tuos pačius keitinius. Dėl (i) priežasties kievienas keitinys buvo pakartotas

$$k_1! \dots k_j! \dots k_n!$$

kartų, o dėl (ii) priežasties –

$$1^{k_1} \dots j^{k_j} \dots n^{k_n}$$

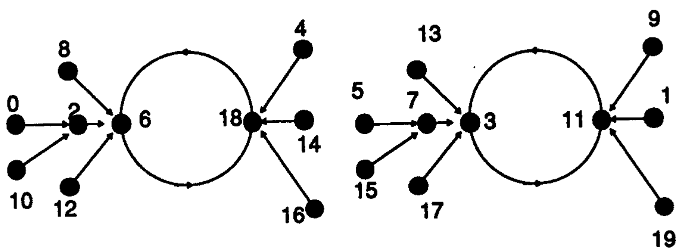
kartų. Padaliję $n!$ iš šių sandaugų, gauname reikiamą formulę. ◇

8. Visi baigtinės aibės atvaizdžiai

Nagrinėsime visų n aibės $X = \{1, \dots, n\}$ atvaizdžių f į ją pačią aibę $\mathbf{T}_n$. Atvaizdžių sąsūkos atžvilgiu ji sudaro pusgrupį, dažnai vadinamą *simetriniu.* Kaip ir bijekcijų atveju f galime apibrėžti lentele, pavyzdžiui,

$$f = \begin{pmatrix} 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \\ 1, 2, 2, 3, 3, 4, 1, 6, 9, 8 \end{pmatrix},$$

nurodančia, kad $f(1) = 1$, $f(2) = 2, f(3) = 2$ ir t.t. Galima vaizduoti ir digrafu, vadinamu *funkciniu* atvaizdžio digrafu (grafu). Atveju $f(x) = x^2 + 2\,(\mathrm{mod} 20)$ turime paveikslą:



Jame dvidešimties taškų aibė atvaizduota į ją pačią. Panašiai, kiekvieną iš n galios aibės atvaizdžių f į ją pačią, galime pavaizduoti numeruotuoju digrafu, kurio viršūnių aibė sutampa su aibės X elementais, o briauna (i, j) yra išvesta iš i į j tada ir tik tada, jei $j = f(i)$. Funkcinį grafą determinuojanti savybė galėtų būti formuluojama šitaip: kievienos viršūnės išėjimo laipsnis (iš jos išvestų briaunų skaičius) lygus vienam.

Matome, kad bet kokio funkcinio grafo struktūrą apibrėžia vektorius $\bar{k}(f) = (k_1, \dots, k_n)$, $1k_1 + \cdots + nk_n = n$, kuriame $k_j = k_j(f)$ žymi j eilės jungių grafo komponenčių skaičių.

Iš atvaizdžio pavaizdavimo lentelė matyti, kad visų n aibės atvaizdžių arba funkcinių n eilės grafų skaičius $|\mathbf{T}_n| = n^n$. Tačiau jungių n eilės funkcinių n eilės grafų kiekį C_n surasti gana sunku. Tuo tikslu pasinaudokime eksponentinėmis genruojančiomis funkcijomis (e.g.f.). Pradžioje pastebėkime porą jų savybių.

Tegu

$$A(t) = \sum_{n\geq 0} \frac{a_n t^n}{n!}, \quad B(t) = \sum_{n\geq 0} \frac{a_n t^n}{n!} \quad -$$

sekų $\{a_n\}$ ir $\{b_n\}$ e.g.f. Tada

$$A(t)B(t) = \sum_{n\geq 0} \frac{t^n}{n!}\left(\sum_{k=0}^{n}\binom{n}{k} a_k b_{n-k}\right).$$

Taigi, sandauga $A(t)B(t)$ yra apskliaustųjų sumų, kai $n = 0, 1, \ldots,$ e..g.f. Panašiai, sekos

$$\sum_{k=0}^{n}\binom{n}{k} a_{k+1} b_{n-k} \tag{1}$$

e.g.f. bus $A'(t)B(t)$.

Teorema. *Tarkime, T_{nk} - skaičius n eilės funkcinių grafų, turinčių k jungių komponenčių, $T_{n,0} = 0$, $\pi(n)$ - skaičius jungių n eilės funkcinių grafų,*

$$T_n(t) = \sum_{k=1}^{n} T_{nk} t^k, \quad T_0(t) = 1, \quad \Pi(y) = \sum_{n\geq 1} \frac{\pi(n) y^n}{n!}.$$

Čia t, y - formalūs parametrai. Tada

$$T(t,y) = \sum_{n\geq 0} \frac{T_n(t) y^n}{n!} = \exp\{t\Pi(y)\}.$$

Įrodymas. Raskime rekurentųjį sąryšį tarp $T_{n+1,k}$ ir T_{nk}. Turėdami $(n+1)$-os viršūnės aibę, pastebėkime, jog $n+1$ viršūnė gali būti jungioje komponentėje, kurioje be jos dar yra $j = 0, 1, \ldots, n$ kitų viršūnių. Turime $\binom{n}{j}$ jų parinkimo galimybių. Galime sudaryti $\pi(j+1)$ jungių funkcinių grafų su $n+1$ ir šiomis j viršūnių. Likusios $n-j$ viršūnių nepriklausomai gali būti $T_{n-j,k-1}$ funkciniuose grafuose. Taigi,

$$T_{n+1,k} = \sum_{j=0}^{n}\binom{n}{j}\pi(j+1) T_{n-j,k-1}.$$

Padauginę iš t^k ir sudėję gautąsias lygybes pagal $k = 1, \ldots, n+1$, turime

$$T_{n+1}(t) = t\sum_{j=0}^{n}\binom{n}{j}\pi(j+1) T_{n-j}(t).$$

Pagal (1) formulę

$$\sum_{n\geq 0}\frac{T_{n+1}(t)y^n}{n!}=T'_y(t,y)=t\bigg(\sum_{n\geq 0}\frac{\pi(n+1)y^n}{n!}\bigg)\bigg(\sum_{n\geq 0}\frac{T_n(t)y^n}{n!}\bigg).$$

Vadinasi,

$$T'_y(t,y)=t\Pi'(y)T(t,y).$$

Integruodami pagal y baigiame teoremos įrodymą. $\diamond$

Išvada. *Teisingas sąryšis*

$$T(y):=T(1,y)=\sum_{n\geq 0}\frac{n^n y^n}{n!}=\exp\{\Pi(y)\}. \tag{2}$$

Iš šioje išvadoje gautojo sąryšio jau būtų galima išvesti funkcijos $\Pi(y)$ Tayloro koeficientų $\pi(n)$ formulę, bet lengviau tą padaryti pasitelkus sekančio skyrelio medžiagą.

9. Numeruotosios kombinatorinės struktūros

Dabar susipažinsime su abstraktesne apibrėžimų sistema, naudojama kombinatorinių struktūrų teorijoje. Galima įsivaizduoti, kad pradedama nuo komponenčių arba nuo jungių kombinatorinių struktūrų, nors toliau įvedamos sąvokos jungumo savybės nereikalauja.

Struktūros yra sudaromos iš elementų (atomų), nurodant jų vidinius ryšius. *Žymėtosiose* struktūrose tiems elementams priskiriami indeksai, dažniausiai skaičiai. Pastaruoju atveju struktūras vadinsime *numeruotosiomis.* Dvi tokios struktūros yra laikomos vienodomis, jei jų elementų numeracijai naudojami tie patys skaičiai ir, sutapatinus vienodai sunumeruotus elementus, vidiniai ryšiai sutampa. Skirtingai apibrėžiant vidinius ryšius tarp elementų, gaunamos skirtingos struktūrų klasės. Fiksuokime vieną tokią klasę $\mathcal{U}$ ir reikalaukime, kad kiekvienam $n\geq 0$ iš n elementų yra sudaroma tik baigtinis skaičius struktūrų, kurių eile (toliau žymėsime $|\cdot|$) laikysime n. Paprastai atvejis $n=0$ atitinka vieną tuščiąją struktūrą, kuri įjungiama į nagrinėjamą klasę arba ne. $n\geq 1$ eilės struktūros elementų numeracijai naudosime tik skaičius $\{1,\dots,n\}$. Visą n eilės struktūrų aibę žymėsime $\mathcal{U}_n\subset\mathcal{U}$, o $u_n=|\mathcal{U}_n|$ – jos elementų skaičių. Pagal susitarimą $u_0\in\{0,1\}$. Taigi, klasę sudaro nesikertančių poaibiŲ sąjunga

$$\mathcal{U}=\bigcup_{n=0}^{\infty}\mathcal{U}_n,\quad u_n<\infty.$$

Formali eilutė

$$U(t)=\sum_{u\in\mathcal{U}}\frac{t^{|u|}}{|u|!}=\sum_{n=0}^{\infty}\frac{u_n t^n}{n!}$$

vadinama ne tik sekos $\{u_n\}$, $n\geq 0$, bet ir klasės $\mathcal{U}$ *eksponentine generuojančia funkcija* (toliau EGF).

Turėdami dvi numeruotų struktūrų klases $\mathcal{U}$ ir $\mathcal{V}$ ir apibrėšime trečią $\mathcal{W}$, vadinamą *skaidumo sandauga.* Ją sudaro visos *sutvarkytosios* poros $w = (u, v) \in \mathcal{U} \times \mathcal{V}$ su visais galimais žemiau aprašytais elementų sunumeravimais. Jei $u \in \mathcal{U}$ elementai buvo numeruoti skaičiais $\{1, \dots, m\}$, o $v \in \mathcal{V}$ – skaičiais $\{1, \dots, n\}$, tai w numeracijai naudojami skaičiai $\{1, \dots, m+n\}$, naujoji struktūra w laikoma $n+m$ eilės. Struktūrų u ir v elementai pernumeruojami, dabar naudojant skaičius $\{1, \dots, m+n\}$, išlaikant buvusį jų sutvarkymą (eiliškumą) ir taip, kad u ir v elementų naujos numeracijos nesikirstų. Formaliai kalbant, skaidumo sandaugos w numeraciją apibrėžia bet kokios dvi monotoniškai didėjančios funkcijos $\theta_1 : \{1, \dots, m\} \to \{1, \dots, m+n\}$ ir $\theta_2 : \{1, \dots, m\} \to \{1, \dots, m+n\}$, kurių reikšmių sritys nesikerta, o jų sąjunga yra visa aibė $\{1, \dots, m+n\}$. Taigi u ir v skaidumo sandauga (ją žymėsim $w = u * v$) yra aibė sutvarkytųjų porų, besiskiriančių pernumeravimu. Aišku, kad būtent θ_i, $i = 1, 2$ monotoniškumas užtikrina ankščiau turėtą struktūrų u ir v elementų sutvarkymą, be to, daugindami skirtingas poras gausime skirtingas skaidumo sandaugas. Visos skaidumo sandaugos $w = u * v$, $u \in \mathcal{U}$, $v \in \mathcal{V}$ sudaro klasių $\mathcal{U}$ ir $\mathcal{V}$ skaidumo sandaugą, kurią žymėsime $\mathcal{W} = \mathcal{U} * \mathcal{V}$. Pagal indukciją apibrėžiama ir bet kokio skaičiaus struktūrų bei jų klasių sandaugos. Atkreipkime dėmesį, kad pradėjome nuo sutvarkytųjų porų (u, v), todėl griežtai kalbant, $\mathcal{U} * \mathcal{V} \neq \mathcal{V} * \mathcal{U}$. Toliau žymėkime

$$\mathcal{U}^{<1>} = \mathcal{U},\ \mathcal{U}^{<2>} = \mathcal{U} * \mathcal{U},\ \dots \mathcal{U}^{<n>} = \mathcal{U} * \mathcal{U}^{<n-1>}, \dots.$$

Pradedant nuo nesutvarkytųjų porų, lygiai taip pat apibrėžiama *Abelio skaidumo sandaugas.* Jų žymėjimui naudosime simbolį $[*]$. Dabar

$$\mathcal{U}^{[1]} = \mathcal{U},\ \mathcal{U}^{[2]} = \mathcal{U}[*]\mathcal{U},\ \dots,\ \mathcal{U}^{[n]} = \mathcal{U}[*]\mathcal{U}^{[n-1]}, \dots$$

Kadangi yra $n!$ kėlinių, sandaugų $\mathcal{U}^{<n>}$ ir $\mathcal{U}^{[n]}$ EGF riša lygybės

$$U^{<n>}(t) = n! U^{[n]}(t), \quad n = 0, 1, \dots. \tag{1}$$

Skaidumo kompleksu vadinsime aibę

$$\mathcal{U}^{<*>} = \{\emptyset\} \cup \mathcal{U} \cup \mathcal{U}^{<2>} \cup \dots. \tag{2}$$

Tai nesikertančių aibių sąjunga. Panašiai

$$\mathcal{U}^{[*]} = \{\emptyset\} \cup \mathcal{U} \cup \mathcal{U}^{[2]} \cup \dots \tag{3}$$

vadinsime *Abelio skaidumo kompleksu* arba *ansambliu.*

Šakniniai numeruotieji miškai bei funkciniai grafai sudaro ansamblius. Pirmuoju atveju pradinė struktūrų klasė buvo visų numeruotų medžių klasė, o antruoju – jungių funkcinių digrafų klasė. Pokštieji miškai sudaro skaidumo kompleksą (ne Abelio), nes juose į medžių tvarką yra atsižvelgiama.

1 teorema. *Tegu*

$$U(t) = \sum_{n \geq 1} \frac{u_n t^n}{n!}, \quad V(t) = \sum_{n \geq 1} \frac{v_n t^n}{n!} \quad -$$

kombinatorinių struktūrų klasių $\mathcal{U}$ bei $\mathcal{V}$ eksponentinės generuojančios funkcijos (EGF). Skaidumo sandaugos $\mathcal{W} = \mathcal{U} * \mathcal{V}$ EGF

$$W(t) = \sum_{n\geq 1} \frac{w_n t^n}{n!} = U(t)V(t). \tag{4}$$

Skaidumo komplekso $\mathcal{U}^{<*>}$ EGF lygi

$$U^{<*>}(t) := \sum_{w\in\mathcal{U}^{<*>}} \frac{t^{|w|}}{|w|!} = (1 - U(t))^{-1}, \tag{5}$$

o Abelio skaidumo komplekso $\mathcal{U}^{[*]}$ EGF –

$$U^{[*]}(t) := \sum_{w\in\mathcal{U}^{[*]}} \frac{t^{|w|}}{|w|!} = e^{U(t)}. \tag{6}$$

Įrodymas. Pastebėkime, kad n eilės skaidumo sandaugų $w = u * v$ galime sudaryti

$$w_n = \sum_{k=0}^{n} \binom{n}{k} u_k v_{n-k},$$

nes pastaroji lygybė nurodo, kad fiksuotoje sandaugoje viena komponentė yra k, o kita – $(n-k)$ eilės, be to, pirmoji komponentė yra numeruota bet kokiu k indeksų poaibiu iš $\{1, \ldots, n\}$. Taigi,

$$\frac{w_n}{n!} = \sum_{k=0}^{n} \frac{u_k}{k!} \, \frac{v_{n-k}}{(n-k)!}.$$

Iš čia išplaukia (4) formulė.

Pasinaudoję ja bei (2) lygybe, gauname

$$U^{<*>}(t) = \sum_{n\geq 0} \sum_{w\in\mathcal{U}^{<n>}} \frac{t^{|w|}}{|w|!} = \sum_{n\geq 0} U(t)^n = (1 - U(t))^{-1}.$$

Abelio skaidumo kompleksui, pasinaudoję (1), turime bei

$$U^{[*]}(t) = \sum_{n\geq 0} \sum_{w\in\mathcal{U}^{[n]}} \frac{t^{|w|}}{|w|!} = \sum_{n\geq 0} \frac{1}{n!} \sum_{w\in\mathcal{U}^{<n>}} \frac{t^{|w|}}{|w|!} = \sum_{n\geq 0} \frac{1}{n!} F(t)^n = e^{U(t)}.$$

◇

Palyginkime gautą rezultatą su 8 skyrelio teoremos išvada. Ji teigia, kad funkcinių digrafų klasės EGF tenkina lygybę

$$T(y) = \sum_{n\geq 0} \frac{n^n}{n!} y^n = \exp\left\{ \sum_{n\geq 1} \frac{\pi(n)}{n!} y^n \right\},$$

čia $\pi(n)$ – jungių funkcinių digrafų skaičius. Kadangi funkciniai digrafai yra jungių digrafų nesutvarkytieji rinkiniai ir funkciniai digrafai yra jungių digrafų generuotas ansamblis, pastarasis sąryšis yra 1 teoremos išvada.

Panagrinėkime kitą pavyzdį. Tegu $\mathcal{U}$ keitinių ciklų klasė. Turėdami $n \geq 1$ skaičių $\{1, \ldots, n\}$ galime sudaryti $n!$ kėlinių $(i_1, i_2, \ldots, i_{n-1}, i_n)$. Kadangi ciklams galioja

$$(i_1, i_2, \ldots, i_{n-1}, i_n) = (i_2, \ldots, i_{n-1}, i_n, i_1) = \cdots = (i_n, i_1, \ldots, i_{n-1}),$$

gausime $(n-1)!$ ciklų. Vadinasi, tokių ciklų klasės EGF lygi

$$\sum_{n \geq 1} \frac{(n-1)!}{n!} t^n = \log(1-t)^{-1}.$$

Abelio skaidumo sandauga $\mathcal{U}^{[n]}$ duotų visus keitinius, sudarytus iš n ciklų. Jos EGF lygi

$$U^{[n]}(t) = \frac{1}{n!} \log^n (1-t)^{-1}.$$

Keitiniai sudarytų ciklų klasės generuotą ansamblį, todėl jo EGF

$$\mathcal{U}^{<*>} = \sum_{n \geq 0} \frac{1}{n!} \log^n (1-t)^{-1} = \frac{1}{1-t},$$

ką mes turėjome ir anksčiau.

Ciklus galime sudarinėti ir iš kitokių negu skaičiai kombinatorinių struktūrų, pvz., medžių. Kokia bus EGF, jei pradėsime nuo struktūrų klasės su EGF $A(t)$?

2 teorema. *Iš kombinatorinių struktūrų klasės $\mathcal{A}$ su EGF $A(t)$ sudarytų ciklų klasės EGF bus lygi*

$$C(t) = \sum_{n \geq 0} \frac{c_n}{n!} t^n = \log \big(1 - A(t)\big)^{-1}.$$

Įrodymas. Kaip matėme anksčiau, n eilės sutvarkytų rinkinių, daromų iš $\mathcal{A}$, EGF lygi $A^n(t)$; ciklams kiekvienas jos koeficientas yra n kartų mažesnis. Tad,

$$C(t) = A(t) + \frac{A^2(t)}{2} + \cdots + \frac{A^n(t)}{n} + \cdots .$$

Tai ir yra 2 teoremos tvirtinimas. $\diamond$

Iš 1 ir 2 teoremų išplaukia įdomių kompleksų generuojančių funkcijų savybių.

1 išvada. *Tegu*

$$D(t) := \sum_{n \geq 1} \frac{d_n t^n}{n!} = \sum_{n \geq 1} \frac{n^{n-1} t^n}{n!} \quad -$$

Cayley'io medžių EGF. Tada $D(t) = te^{D(t)}$, kai $|t| < e^{-1}$.

Įrodymas. Pasinaudoję Stirlingo formule, nesunkiai nustatome eilutės $D(t)$ konvergavimo sritį $|t| < e^{-1}$. Pastebime, kad šakniniai miškai sudaro Cayley'io medžių ansamblį. Vadinasi, pagal 1 teoremą jų EGF išsireiškia per $D(t)$. Gauname

$$Q(t) := \sum_{n\geq 0} \frac{q_n t^n}{n!} = e^{D(t)}.$$

Pagal 6.2 teoremą $q_n = d_{n+1}/(n+1)$, todėl

$$Q(t) = \sum_{n\geq 0} \frac{d_{n+1} t^n}{(n+1)!} = t^{-1} D(t).$$

◇

2 išvada. *Tegu* $\Pi(t)$ *– jungių funkcinių digrafų EGF, o* $T(t)$ *– visų funkcinių digrafų ansamblio EGF. Tai srityje* $|t| < e^{-1}$

$$\Pi(t) = \log\big(1 - D(t)\big)^{-1}, \qquad T(t) = \frac{1}{1 - D(t)}.$$

Įrodymas. Kiekviena funkcinio digrafo komponentė yra sudaryta iš ciklo ir Cayley'io medžių, kurių briaunos šįkart turi kryptis nukreiptas į šaknis, esančias šiame cikle. Medžiai gali būti ir pirmos eilės, tai bus ciklo viršūnės. Ciklas apibrėžia ir medžių išdėstymo tvarką, sukeitus bent du iš jų, gaunamas kito atvaizdžio digrafas. Kitaip tariant, į jungią funkcinio digrafo komponentę galime žiūrėti kaip į medžių ciklą. Jei $\pi(n)$ – n eilės jungių funkcinių digrafų skaičius, $n \geq 1$, tai šis skaičius reikš ir tos pačios eilės medžių ciklų kiekį. Pagal 2 teoremą medžių ciklų klasės EGF lygi

$$\log\big(1 - D(t)\big)^{-1}.$$

Kadangi $\mathcal{T}$ yra šios klasės generuotas ansamblis, pagal 1 teoremą gauname

$$T(t) = \exp\big\{\log\big(1 - D(t)\big)^{-1}\big\} = \big(1 - D(t)\big)^{-1}. \tag{6}$$

Išvada įrodyta. ◇

Naudojant 1 ir 2 išvadas ir šią lygybę nebesunku rasti $\pi(n)$ bei jo asimptotiką, kai $n \to \infty$.

Lagrange lema. *Tegu funkcija* $f(z)$ *yra netiesiogiai apibrėžta lygybės*

$$f(z) = z\phi(f(z)) \tag{7}$$

pagalba, kai $\phi(u)$ *– analizinė taško* $u = 0$ *aplinkoje ir* $\phi(0) = 1$*. Jei* $g(z)$ *yra analizinė taško* $z = 0$ *aplinkoje, tai sudėtinė funkcija* $g(f(z))$ *irgi yra analizinė kažkokioje nulinio taško aplinkoje, be to, jos* n*-asis Tayloro koeficientas lygus funkcijos* $\phi(u)^n g'(u)$ $(n-1)$*-am Tayloro koeficientui* c_{n-1}*, padalytam iš* n*.*

Įrodymas. Tegu $u = f(z)$ ir $z = z(u)$ – jai atvirkštinė funkcija. Iš (7) turime, kad $z = u/\phi(u)$. Pagal lemos sąlygas abi yra analizinės tam tikrose nulinių taškų aplinkose. Todėl naudodami Koši formulę su pakankamai mažais $\rho, \rho_1 > 0$, gauname

$$
\begin{aligned}
c_{n-1} &= \frac{1}{2\pi i}\int_{|u|=\rho}\frac{\phi(u)^n g'(u)}{u^n}\,du = \frac{1}{2\pi i}\int_{|u|=\rho}\frac{g'(u)}{z(u)^n}\,du = \\
&= \frac{1}{2\pi i}\int_{|z|=\rho_1}\frac{g'(f(z))f'(z)}{z^n}\,dz = \frac{1}{2\pi i}\int_{|z|=\rho_1}\frac{d\big(g(f(z))\big)}{z^n} = \\
&= \frac{n}{2\pi i}\int_{|z|=\rho_1}\frac{g(f(z))\,dz}{z^{n+1}}.
\end{aligned}
$$

Įžiūrėję pastarojo integralo prasmę, baigiame lemos įrodymą. $\diamond$

3 teorema. *Jungių $n \geq 1$ eilės funkcinių grafų skaičius*

$$\pi(n) = (n-1)!\sum_{k=0}^{n-1}\frac{n^k}{k!} = n!e^n\bigg(\frac{1}{2n} + O(n^{-3/2})\bigg),$$

Įrodymas. Teoremos 1 pirmoje išvadoje gavome $D(z) = ze^{D(z)}$. Pagal (6) reikia rasti n-ą funkcijos $\log(1-D(z))^{-1}$ Tayloro koeficientą ir padauginti jį iš $n!$. Pasinaudojame Lagrange lema, kai $\phi(u) = e^u$, o $g(u) = \log(1-u)^{-1}$, ir gauname $\phi^n(u)g'(u) = e^{nu}/(1-u)$. Nesunkiai randame $(n-1)$-ą Tayloro koeficientą. Jis lygus $\sum_{0\leq k\leq n-1} n^k/k!$. Padalię iš n ir padauginę iš $n!$, randame $\pi(n)$ išraišką.

Sumos aproksimavimui pasinaudokime Bery-Eseno teorema apie konvergavimo greitį centrinėje ribinėje teoremoje. Tegu $S_n = Z_1 + ... + Z_n$ – nepriklausomų vienodai pasiskirsčiusių Poissono dydžių su vienetiniu parametru ($\mathbf{E}Z_j = 1$) suma. Todėl S_n irgi Poissono dydis su parametru n, o

$$
\begin{aligned}
e^{-n}\sum_{k=0}^{n-1}\frac{n^k}{k!} &= P(S_n \leq n-1) = P((S_n - n)/\sqrt{n} \leq -1/\sqrt{n}) = \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{-1/\sqrt{n}} e^{-u^2/2}\,du + O(n^{-1/2}) = \frac{1}{2} + O(n^{-1/2}).
\end{aligned}
$$

Įstatę šį įvertį į $\pi(n)$ išraišką, baigiame teoremos įrodymą. $\diamond$

Palyginimui pastebėkime, kad jungiųjų n eilės funkcinių digrafų ir visų tokios eilės digrafų santykis

$$\frac{\pi(n)}{n^n} \sim \sqrt{\frac{\pi}{2n}}, \quad n \to \infty.$$

Panašiai išvystoma ir nenumeruotųjų kombinatorinių struktūrų teorija.

10. Nenumeruotųjų struktūrų kompleksai

Nagrinėti nenumeruotų struktūrų pavyzdžiai turi bendrą bruožą: sudėtingesni objektai yra sudaryti iš atskirų dalių. Polinomus sudaro pirminiai daugikliai, binariuosius medžius - pomedžiai, į kurių tvarką yra atsižvelgiama . Atskiros dalys gali būti net lygios. Nagrinėjamus objektus, turinčius apibrėžtą natūralųjį skaičių *svorį* (atskirais atvejais tai gali būti laipsnis, eilė ir pan.), vadinkime *svorinėmis kombinatorinėmis struktūromis.* Tegu $\mathcal{P}$ yra tam tikra kombinatorinių struktūrų klasė, $\kappa \in \mathcal{P}$ viena struktūra, o $w(\kappa)$ - jos svoris. Reikalaukime, kad klasėje $\mathcal{P}$ yra tik baigtinis n svorio struktūrų skaičius

$$\pi_n = |\{\kappa \in \mathcal{P} : \ w(\kappa) = n\}|, \quad n \geq 1.$$

Bet koks rinkinys

$$\sigma := \{\kappa_1, \dots, \kappa_s\}, \quad \kappa_i \in \mathcal{P}, \quad 1 \leq i \leq s,$$

gal būt pasikartojančių elementų gali būti laikomas nauja kombinatorine struktūra. Tuo tikslu, reikia suteikti jai svorį. Natūralu jį apibrėžti lygybe

$$w(\sigma) = w(\kappa_1) + \cdots + w(\kappa_s).$$

Tuščiajam rinkiniui $\sigma = \emptyset$ suteikime nulinį svorį. Taip apibrėžtos struktūros σ vadinamos *kartotinėmis aibėmis* (*multiaibėmis*), o jų visuma, įskaitant ir tuščią, - aibės $\mathcal{P}$ *kartotinių poaibių* (*multiaibių*) struktūra. Ją žymėkime $K(\mathcal{P})$.

Pirminių polinomų, kurių vyriausias koeficientas lygus vienam, virš baigtinio kūno rinkinys yra geriausias tokios kartotinės struktūros pavyzdys. Sutapatinę ją su rinkinio polinomų sandauga, visų polinomų aibę galėtume laikyti kartotine pirminių polinomų struktūra. Aišku, kad svorių vaidmenį vaidina polinomų laipsniai; $\pi_n = \pi(n)$ - skaičių n-ojo laipsnio pirminių polinomų - nagrinėjome 3 skyrelyje.

Žvelgdami į natūraliojo skaičiaus adityviojo skaidinio dėmenis kaip į natūraliųjų skaičių aibės kartotinį poaibį, tokius skaidinius irgi galėtume vadinti kartotinių poaibių struktūra, kurioje svorio vaidmenį vaidina natūraliųjų skaičių didumai. Dabar $\pi_k = 1$ su kiekvienu $k \in \mathbf{N}$.

Pradėdami nuo $\mathcal{P}$ poaibių, (dabar pasikartojančių elementų κ_i neimtume), panašiai gautume aibės $\mathcal{P}$ *poaibių struktūrą*. Ją žymėkime $P(\mathcal{P})$.

Formalias laipsnines eilutes

$$\Pi(z) = \sum_{\kappa \in \mathcal{P}} z^{w(\kappa)} = \sum_{n=0} \left(\sum_{w(\kappa)=n} 1 \right) z^n =: \sum_{n=0} \pi_n t^n,$$

$$K(z) = \sum_{\sigma \in K(\mathcal{P})} z^{w(\sigma)} = \sum_{n=0} \left(\sum_{w(\sigma)=n} 1 \right) z^n =: \sum_{n=0} k_n t^n$$

ir

$$P(z) = \sum_{\sigma \in P(\mathcal{P})} z^{w(\sigma)} = \sum_{n=0} \left(\sum_{w(\sigma)=n} 1 \right) z^n =: \sum_{n=0} p_n z^n.$$

vadinkime atitinkamų struktūrų $\mathcal{P}$, $K(\mathcal{P})$ ir $P(\mathcal{P})$ arba sekų $\{\pi_n\}$, $\{k_n\}$ bei $\{p_n\}$ *generuojančiomis funkcijomis.* Raskime jų sąryšius.

1 teorema. *Teisingos formalios lygybės*:

$$K(z) = \exp\left\{\sum_{m=1}^{\infty} \frac{\Pi(z^m)}{m}\right\},$$

$$P(z) = \exp\left\{\sum_{m=1}^{\infty} \frac{(-1)^m \Pi(z^m)}{m}\right\}.$$

Įrodymas. Kaip ir 3 skyrelyje gauname

$$k_n = \sum_{\substack{\bar{k} \in \mathbf{Z}^{+n} \\ 1k_1 + \cdots + nk_n = n}} \prod_{j=1}^{n} \binom{\pi_j + k_j - 1}{k_j}.$$

Vadinasi, pakartoję ankstesnius skaičiavimus, įrodytume lygybę

$$K(z) = \sum_{n \geq 0} k_n z^n = \prod_{j \geq 1} (1 - z^j)^{-\pi_j}.$$

Toliau panaudodami logaritminės funkcijos skleidimo Tayloro eilute formulę, gauname

$$K(z) = \exp\left\{\sum_{j \geq 1} \pi_j \sum_{m \geq 1} \frac{z^{mj}}{k}\right\} = \exp\left\{\sum_{m \geq 1} \frac{1}{m} \Pi(z^m)\right\}.$$

Antrosios įrodymui pastebėkime, kad

$$p_n = \sum_{\substack{\bar{k} \in \mathbf{Z}^{+n} \\ 1k_1 + \cdots + nk_n = n}} \prod_{j=1}^{n} \binom{\pi_j}{k_j}.$$

Be to,

$$\prod_{j \geq 1} (1 + z^j)^{\pi_j} = \prod_{j \geq 1} \sum_{k=0}^{\infty} \binom{\pi_j}{k} z^{kj} = \sum_{n=0}^{\infty} z^n \sum_{\substack{\bar{k} \in \mathbf{Z}^{+n} \\ 1k_1 + \cdots + nk_n = n}} \prod_{j=1}^{n} \binom{\pi_j}{k_j} = P(z).$$

Logaritmuodami sandaugą, kaip ir anksčiau iš čia gautume antrąją lemos lygybę. ◇

Turėdami keletą skirtingų pradinių struktūrų klasių $\mathcal{P}_k$, $k \geq 2$, galėtume sudaryti dar įdomesnių naujų struktūrų. Dabar apibrėšime sekų struktūrą. Tegu $\mathcal{P}'$ ir $\mathcal{P}''$ dvi struktūros. *Sutvarkytųjų porų struktūra* vadinsime $\mathcal{P}' \times \mathcal{P}''$, kurioje poros $\sigma := (\kappa', \kappa'')$ svoriu laikoma $w(\sigma) = w(\kappa') + w(\kappa'')$. Panašiai apibrėšime ir $\mathcal{P}$ laipsnius.

1 teorema. *Jei π'_k ir π''_k yra k-ojo svorio struktūrų aibėse $\mathcal{P}'$ ir $\mathcal{P}''$ skaičiai, tai n-jo svorio sutvarkytųjų porų struktūrų yra*

$$\sum_{k=1}^{n-1} \pi'_k \pi''_{n-k}.$$

Įrodymas. Išplaukia iš apibrėžimų. ◇

Aibės

$$\{\emptyset\} \cup \mathcal{P} \cup \mathcal{P}^2 \cup \dots$$

elementai vadinami $\mathcal{P}$ *sekų struktūromis.* Žymėkime ją $S(\mathcal{P})$.

2 teorema. *Sekų struktūros generuojanti funkcija lygi*

$$\sum_{\sigma \in S(\mathcal{P})} z^{w(\sigma)} = 1 + \Pi(z) + \Pi(z)^2 + \dots = \frac{1}{1 - \Pi(z)}.$$

Įrodymas. Išplaukia iš apibrėžimų. ◇

Dar kartą prisiminkime binariuosius medžius ir Katalano skaičius. Kadangi kiekvienas binarusis medis T yra arba viena išorinė viršūnė $\circ$, arba vidinė viršūnė $*$ ir dviejų binariųjų medžių seka, todėl gauname tokią formalią schemą:

$$\{T\} = \{\circ\} \cup \{(*, T, T)\}.$$

Čia paskutinė aibė yra sudaryta iš sekų. Priskirdami išorinėms viršūnėms vienetinius svorius, o vidinėms viršūnėms - nulius, gautume nagrinėtas struktūras. Užrašę atitinkamas generuojančias funkcijas, gauname lygybę

$$C(z) := \sum_{n=1}^{\infty} C_n z^n = z + 1 \cdot C(z) \cdot C(z) = z + C^2(z),$$

kurią jau buvome 4 skyrelyje.

Kaip ir praeitame skyrelyje galėtume apibrėšti ir nenumeruotų struktūrą ciklų klasę. Reziumuodami akcentuosime, kad formalus naujų struktūrų klasių sudarymas duoda ir jų generuojančių funkcijų ryšius. Panaudodami pastaruosius, galime naujas struktūras suskaičiuoti.