Dalyko „Algoritmai ir duomenų struktūros“ 2021 m. egzamino užduočių sprendimai

2021-06-22

1. Algoritmų, kuriais atspausdinama $n \in \mathbb{N}$ pirmųjų tribonačio sekos narių, gali būti labai įvairių. Žemiau pateiktas pavyzdys, kuriame programa išveda sąrašus [0] arba [0, 0], kai atitinkamai $n = 1$ arba $n = 2$. Kitu atveju sudaromas sąrašas $t \leftarrow [0, 0, 1]$, kurio gale įterpiama $n - 3$ naujų elementų, lygių paskutinių 3 sąrašo elementų sumai, ir išvedamas sąrašas t.

```
tribonacci (n):
    if n = 1 then
        return([0])
    else if  n = 2 then
        return([0, 0])
    else
        t ← [0, 0, 1]
        for i ← 2 to n − 2 do
            t.append(t[i] + t[i − 1] + t[i − 2])
        return(t)
```

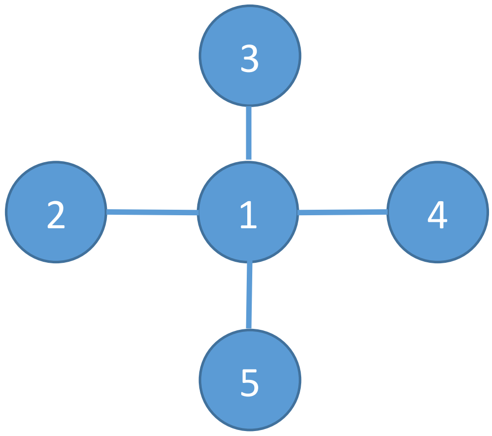
2. Pasinaudosime šiomis Eratosteno rėčio algoritmo savybėmis:
    1. Vienintelis lyginis pirminis skaičius yra 2 (kiti lyginiai išbraukiami).
    2. Vienintelis trijų kartotinis pirminis skaičius yra 3 (kiti išbraukiami).

   Tarkime, kad pradinis ieškomos natūraliųjų skaičių sekos elementas yra $n > 4$. Tada seka turi pradėti ir baigtis nelyginiu skaičiumi, kad joje pirminių skaičių būtų daugiau nei sudėtinių. Bet tokiu atveju jokie iš eilės einantys trys nelyginiai skaičiai negali būti pirminiai, nes kažkuris bus dalus iš 3. Išvada: kai $n > 4$, daugiausiai iš eilės einančių natūraliųjų skaičių ieškomoje sekoje gali būti tik 3. Kai $n \leq 4$, tokią seką nesunku sudaryti: [1, 2, 3, 4, 5, 6, 7, 8], kurioje 4 pirminiai skaičiai, 3 sudėtiniai ir 1 nei pirminis, nei sudėtinis. Atsakymas: 8 skaičiai.

3. a) Panašumų ir skirtumų tarp paieškos į plotį ir į gylį galima aptikti daug daugiau, išskirti keli esminiai.

<table>
<tr><th>Paieška į plotį</th><th>Paieška į gylį</th></tr>
<tr><td colspan="2">Abu algoritmai skirti apeiti grafo arba digrafo viršūnėms.</td></tr>
<tr><td colspan="2">Abiejuose algoritmuose gali būti naudojami papildomi spalvų atributai viršūnėms: balta spalva žymima neaptikta viršūnė, pilka spalva – neapdorota viršūnė, juoda spalva – apdorota viršūnė.</td></tr>
<tr><td colspan="2">Abiejų algoritmų vienodas sudėtingumas O(|V|+|E|).</td></tr>
<tr><td>Realizacijai patogu naudoti eilutės (queue) duomenų struktūrą.</td><td>Realizacijai patogu naudoti dėklo (dequeue) duomenų struktūrą.</td></tr>
<tr><td>Pirma aptinkamos visos gretimos viršūnės pradinei. Pats algoritmo pavadinimas ir pasako, kad vyksta paieška platyn.</td><td>Paieška vyksta rekursijos principu iš ką tik atrastos viršūnės, dėl to, ieškant neatrastų viršūnių, keliaujama gilyn.</td></tr>
</table>

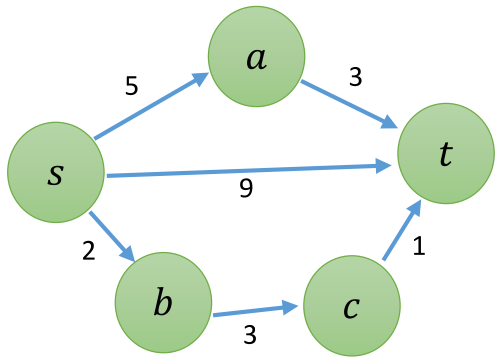
3. b) Žemiau pateiktas jungaus 5 eilės numeruotojo grafo pavyzdys.



Pradedant 1 viršūne, likusios bus apeinamos ta pačia tvarka tiek paieškos į plotį, tiek į gylį algoritmais. Pradedant kuria nors kraštine viršūne, antra atrasta viršūnė bus 1, o likusios ta pačia tvarka bus atrandamos tiek paieškos į plotį, tiek į gylį algoritmais.

4. Dvejetainė paieška vyksta išrikiuotame sąraše. Šios paieškos tikslas – patikrinti, ar sąrašui priklauso kuris nors elementas pagal įvestą reikšmę. Jei ieškoma reikšmė sutampa su vidurinio sąrašo elemento reikšme, algoritmas stabdomas ir išvedamas atsakymas. Kitu atveju algoritmas rekursyviai tęsiamas tame posąrašyje, kuriame gali būti rasta ieškoma elemento reikšmė. Atkreipkime dėmesį, kad pačiu nepalankiausiu atveju, t. y. esant sąrašo ilgiui 2^k, $k \in \mathbb{N}$, algoritmo vykdymo eigoje turėsime 2^k, 2^{k-1}, 2^{k-2}, ... , 2^1, 2^0 ilgio posąrašius, iš kurių posąrašis, kurio ilgis $2^0 = 1$ garantuos, kad reikšmė rasta arba ne. Jei sąrašo ilgis yra $n = 2^k$, tada $k = \log_2 n$, kas ir parodo galimų iteracijų skaičių. Kadangi dvejetainės paieškos kiekviena iteracija yra konstantinio sudėtingumo, tai bendras algoritmo sudėtingumas lygus $O(1) \cdot O(\log n) = O(\log n)$.

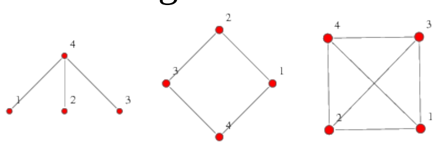
5. Žemiau pateiktas svorinio digrafo pavyzdys.



Iš pradžių atstumas nuo s iki t, (atributas $d[t]$) bus priskirtas begalybei. Tada algoritmo vykdymo eigoje bus atrasti nauji takai nuo s iki t ir atributo $d[t]$ reikšmė sumažinta (pagerinta) 3 kartus:

takas digrafe	$d[t]$
$s \to t$	9
$s \to a \to t$	5+3=8
$s \to b \to c \to t$	2+3+1=6

6. Užduotis gana lengvai suvokiama numeruotąjį (bekilpį) n-tosios eilės grafą susiejus su jį atitinkančia simetrine gretimumo matrica.



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Pateiktame pavyzdyje matome 4 eilės grafus ir juos atitinkančias gretimumo matricas. Reikia rasti visų tokių galimų matricų skaičių, kai grafo eilė lygi n. Kadangi tokios matricos simetrinės, o įstrižainėse nuliai, tai galimų skirtingų reikšmių pirmoje eilutėje yra $n-1$, antroje eilutėje $n-2$ ir paskutinėje 1. Viso tokių reikšmių $1+2+\cdots+n-1=\frac{n(n-1)}{2}$. Kadangi kiekviena tokia reikšmė gali būti 0 arba 1, tai viso turėsime $2^{\frac{n(n-1)}{2}}$ skirtingų matricų ir tuo pačiu n-tosios eilės numeruotųjų bekilpių grafų.

7. Prisiminkime, kad Priuferio kode įrašomi vidinių medžio viršūnių numeriai, tarkime, jų iš viso yra n ir tada medis turi iš viso $n+2$ viršūnių. Belieka atimti iš visų medžio viršūnių vidines viršūnes: $n+2-n=2$. Atsakymas: 2 lapai.