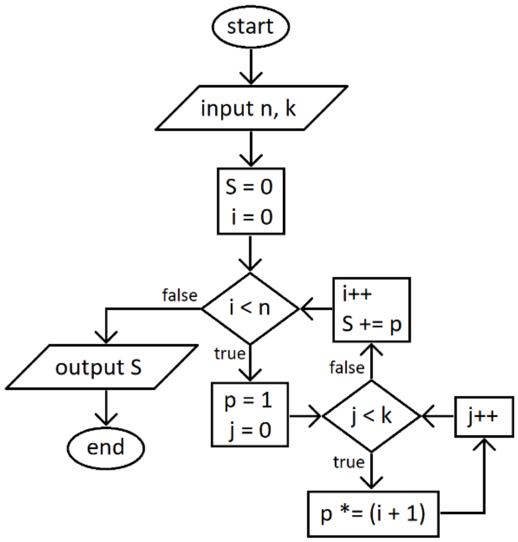
1. Pseudokodą galime pavaizduoti paprasta *Python* programa:

```python
def suma(n, k):
    S = 0
    for i in range(n):
        p = 1
        for j in range(k):
            p *= (i + 1)
        S += p
    return(S)
```

Ją atitinkanti blokinė schema:



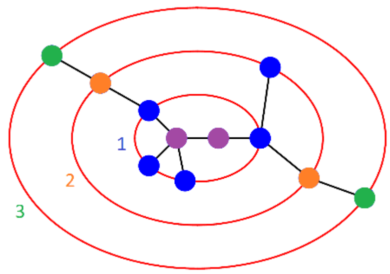
Nesunku pastebėti, kad šio algoritmo 2 cikluose vykdoma atitinkamai n ir k operacijų, kadangi vyksta ciklas cikle, tai algoritmo sudėtingumas O(nk).

2. Pažymėkime keturženklį šešioliktainį skaičių $XYZW_{16}$.
   a) Kiekvieną jo skaitmenį iš aibės {A, B, C, D, E, F} galime parinkti 6 būdais, todėl viso skirtingų tokių skaičių yra $6^4 = 1296$.
   b) Apskaičiuokime visų tokių galimų skaičių sumą dešimtainiu pavidalu. Pažymėkime kiekvieną tokios sumos narį pavidalu $XYZW_{16} = X_{10} \cdot 16^3 + Y_{10} \cdot 16^2 + Z_{10} \cdot 16^1 + W_{10} \cdot 16^0$. Į kiekvieną iš pozicijų X_{10}, Y_{10}, Z_{10}, W_{10} galime parinkti iš aibės {A, B, C, D, E, F} skaičius $6^4 = 1296$ skirtingais būdais, todėl bendra jų suma kiekvienoje iš 4 pozicijų lygi $6^3 \cdot (A + B + C + D + E + F)_{16} = 6^3 \cdot (10 + 11 + 12 + 13 + 14 + 15)_{10} = 216 \cdot 75 = 16200$. Todėl galutinė suma lygi $16200 \cdot 16^3 + 16200 \cdot 16^2 + 16200 \cdot 16^1 + 16200 \cdot 16^0 = 70777800$.

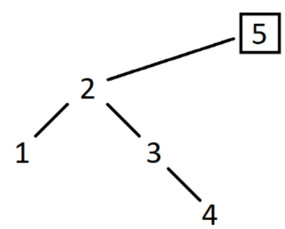
   Pastaba. a) ir b) dalių atsakymus galime patikrinti paprasta *Python* programa:

```python
Q = [10, 11, 12, 13, 14, 15]
answ = 0
answ2 = 0
for X in Q:
    for Y in Q:
        for Z in Q:
            for W in Q:
                answ += 1
                answ2 += X * 16 ** 3 + Y * 16 ** 2 + Z * 16 + W
print(answ)
print(answ2)
```

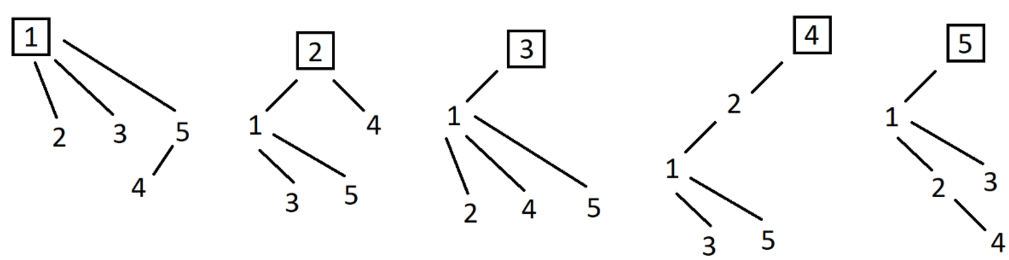
3. Galima daugeliu būdų paaiškinti, kad ilgiausias medžio takas visada eina per to medžio centrą. Kai medis turi vieną arba dvi viršūnes, tai akivaizdu. Tarkime, kad medis turi daugiau nei dvi viršūnes. Pavaizduokime medį radialiniu būdu lygmenimis, kurių numeriai parodo, kokiu atstumu to lygmens viršūnės yra nutolusios nuo medžio centro, kurį sudaro viena ar dvi viršūnės:



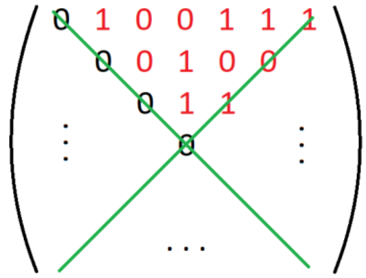
Atkreipkime dėmesį, kad bet kurios to paties lygmens viršūnės nėra sujungtos briauna (kitu atveju medis turėtų ciklą ir nebebūtų medis). Todėl, pašalinus medžio centrą, pirmojo lygmens viršūnės tampa izoliuotos viena nuo kitos, todėl tuo pačiu tampa izoliuotos ir kitos dvi, labiausiai viena nuo kitos nutolusios viršūnės medyje. Išvada: ilgiausias medžio takas visada eina per medžio centrą.
4. Žr. 2020 metų 5 užd. sprendimą ir 2023 metų atsakymus.
5. Jei Priuferio kodą sudaro 3 skaičiai, medis turės 5 viršūnes. Tarkime, kad duotas Priuferio kodas yra korektiškas, t. y. $1 \le a, b, c \le 5$.
   a) Taip sudarytas medis vieninteliu atveju nebus dvejetainis medis, jei $a = b = c$, nes vidinė viršūnė turės 4 gretimas, kas medžio struktūroje neleidžiama.
   b) Priklauso nuo a, b ir c reikšmių. Panagrinėkime kelis atvejus, kai dvejetainis paieškos medis galimas ir kai ne.
      i. Tarkime $a = 2$, $b = 3$ ir $c = 2$. Pagal Priuferio kodą gaunamas medis {{1, 2}, {2, 3}, {2, 5}, {3, 4}}. Jei 5 viršūnę pavaizduotume kaip šakninę, galėtume sudaryti dvejetainį paieškos medį:



      ii. Tarkime $a = 1$, $b = 2$ ir $c = 1$. Pagal Priuferio kodą gaunamas medis {{1, 2}, {1, 3}, {1, 5}, {2, 4}}. Kad ir kokią šakninę viršūnę beparinktume, šis medis niekada nebus dvejetainis paieškos medis:



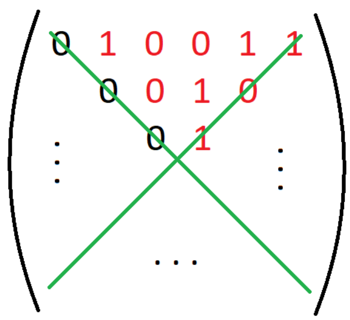
6. Galima išskirti 2 atvejus.
   a) m – nelyginis skaičius. Tokiu atveju simetrinę gretimumo matricą vienareikšmiškai apibrėš pavyzdyje, kai $m = 7$, raudona spalva pažymėti skaičiai iš aibės {0, 1}:



viso galimų tokių skaičių yra

f(m) = (m – 1) + (m – 3) + (m – 5) + ... + 2 = (2 + m – 1) / 2 * (m – 1) / 2 = $(m^2 – 1) / 4$.

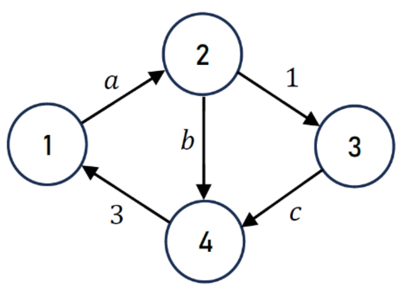
b) m – lyginis skaičius. Tokiu atveju simetrinę gretimumo matricą vienareikšmiškai apibrėš pavyzdyje, kai m = 6, raudona spalva pažymėti skaičiai iš aibės {0, 1}:



viso galimų tokių skaičių yra
f(m) = (m – 1) + (m – 3) + (m – 5) + ... + 1 = (1 + m – 1) / 2 * m / 2 = $m^2 / 4$.

Todėl iš viso skirtingų gretimumo matricų, simetrinių abiejų įstrižainių atžvilgiu, yra $2^{f(m)}$. Pavyzdžiui, kai m = 37, turime 2^342 skirtingų matricų arba kai m = 38, turime 2^361 skirtingų matricų ir t. t.

7. Tarkime, kad a, b ir c yra teigiami skaičiai. Vykdykime Floido-Varšalo algoritmu trumpiausių takų paiešką digrafe



Pirma apskaičiuokime D_0 atstumų matricą, kurioje viršūnė v pasiekiama iš viršūnės u vienos briaunos atstumu (nepasiekiamoms viršūnėms v iš u priskiriamos begalinio atstumo reikšmės).

0	a	∞	∞
∞	0	1	b
∞	∞	0	c
3	∞	∞	0

Atnaujindami D_0 matricą, sudarome D_1 matricą, kurioje viršūnė v pasiekiama iš viršūnės u take, einančiame per viršūnę 1:

0	a	∞	∞
∞	0	1	b
∞	∞	0	c
3	a + 3	∞	0

Atnaujindami D_1 matricą, sudarome D_2 matricą, kurioje viršūnė v pasiekiama iš viršūnės u take, einančiame per viršūnę 2:

0	a	a + 1	a + b
∞	0	1	b
∞	∞	0	c
3	a + 3	a + 4	0

Atnaujindami D_2 matricą, sudarome D_3 matricą, kurioje viršūnė v pasiekiama iš viršūnės u take, einančiame per viršūnę 3:

0	a	a + 1	min(a + b, a + c + 1)
∞	0	1	min(b, c + 1)
∞	∞	0	c
3	a + 3	a + 4	0

Atnaujindami D_3 matricą, sudarome D_4 matricą, kurioje viršūnė v pasiekiama iš viršūnės u take, einančiame per viršūnę 4:

0	a	a + 1	min(a + b, a + c + 1)
min(b + 3, c + 4)	0	1	min(b, c + 1)
c + 3	a + c + 3	0	c
3	a + 3	a + 4	0

Pavyzdžiui, kai a = 1, b = 4, c = 2, atstumų matricos D_0, D_1, D_2, D_3, D_4 lygios:

D_0				**D_1**				**D_2**				**D_3**				**D_4**			
0	1	∞	∞	0	1	∞	∞	0	1	2	5	0	1	2	4	0	1	2	4
∞	0	1	4	∞	0	1	4	∞	0	1	4	∞	0	1	3	6	0	1	3
∞	∞	0	2	∞	∞	0	2	∞	∞	0	2	∞	∞	0	2	5	6	0	2
3	∞	∞	0	3	4	∞	0	3	4	5	0	3	4	5	0	3	4	5	0