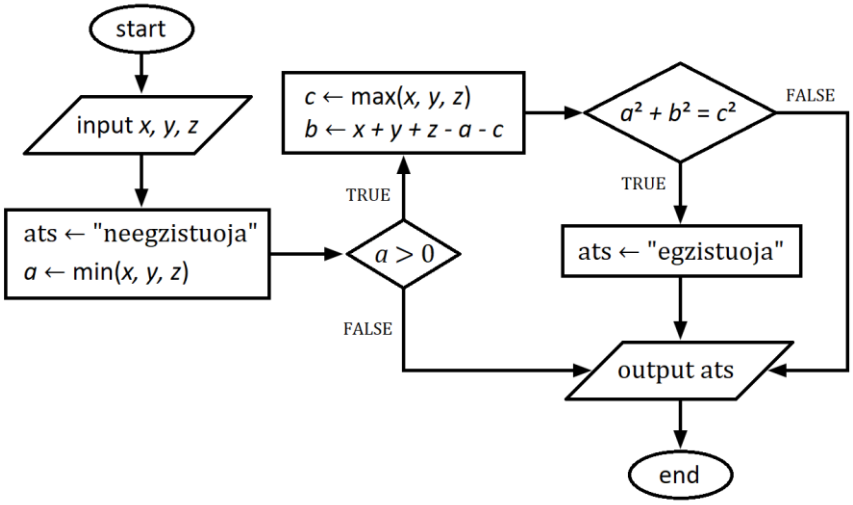
Dalyko „Algoritmai ir duomenų struktūros“ 2022 m. egzamino užduočių sprendimai

2022-06-27

1. Savaime suprantama, kad reikia nustatyti, ar skaičiai x, y, z sudaro „Pitagoro trejetą“. Šiuos skaičius patogu išrikiuoti didėjimo tvarka $a \le b \le c$ ir patikrinti, ar $a^2 + b^2 = c^2$. Taip pat reikia įvertinti, ar įvesties duomenys x, y, z yra teigiami skaičiai (kitu atveju gali atsirasti klaidos, pavyzdžiui, $0^2 + 0^2 = 0^2$, $1^2 + 0^2 = 1^2$ arba $(-3)^2 + (-4)^2 = (-5)^2$).
   a) Vieno iš galimų algoritmų pseudokodas:

```
def ar_statusis(x, y, z):
    ats = "neegzistuoja"
    a = min(x, y, z)
    if a > 0:
        c = max(x, y, z)
        b = x + y + z - a - c
        if a * a + b * b == c * c:
            ats = "egzistuoja"
    return(ats)
```

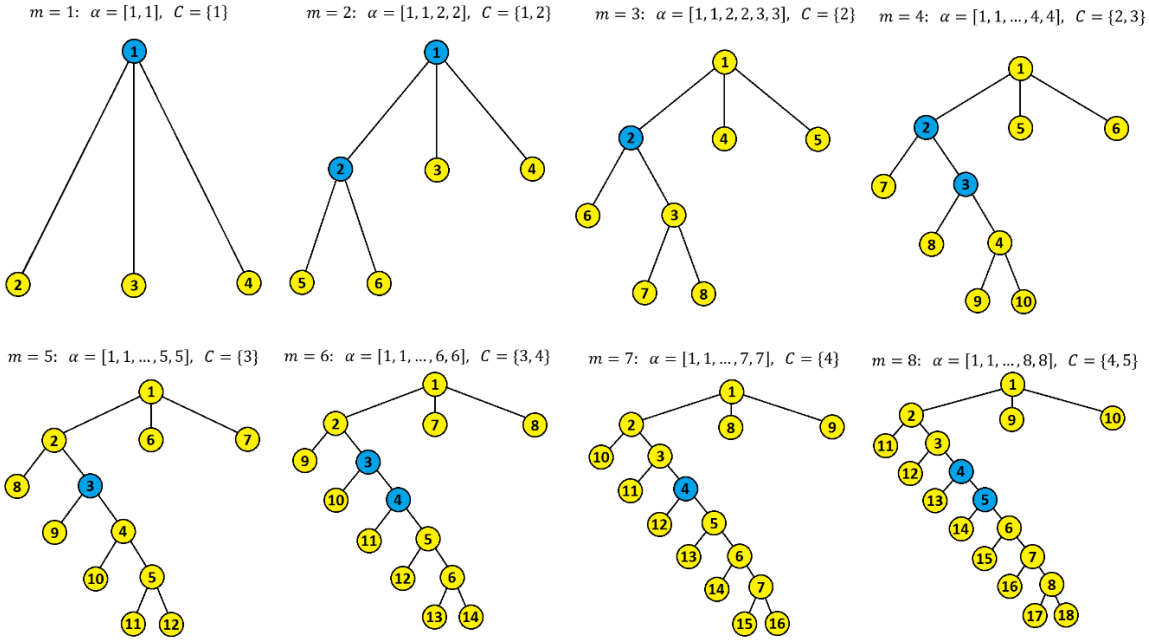
   b) Blokinė schema:



   P. S. galima ir dar paprastesnį algoritmą sudaryti pasinaudojus savybe, kad trikampis yra statusis, jeigu jo kraštinių ilgiai x, y, z tenkina sąlygą $x^2 + y^2 + z^2 = 2 \cdot \max(x, y, z)^2$.

2. Tarkime, kad k yra didžiausias skaičius intervale $[1; k]$, kuriame reikia rasti visus pirminius skaičius taikant Eratosteno rėčio algoritmą. Šio algoritmo vykdymo metu bus išimtinai išbraukiamas vienetas, tada išbraukiami neviršijančių $\sqrt{k}$ pirminių skaičių 2, 3, 5, 7, ... kartotiniai. Tai ekvivalentu patikrinimui, ar k nesidalija (be liekanos) iš visų pirminių skaičių, neviršijančių $\sqrt{k}$.
   Išvada: teiginys yra teisingas visais atvejais, išskyrus, kai $k = 1$, nes teigiant, kad 1 nėra pirminis, seka, kad iš viso nėra pirminių skaičių, neviršijančių $\sqrt{k}$ ir gaunama išvada, kad 1 yra pirminis!

3. Susiekime šį vaikų žaidimą su elemento paieška išrikiuotame sąraše $[1, 2, ..., n]$.
   a) Geriausia strategija yra kiekvieno spėjimo metu rinktis vidurinį išrikiuoto sąrašio elementą ir po to kartoti šią strategiją atsižvelgiant į užuominas „skaičius didesnis“ arba „skaičius mažesnis“. Tai atitinka dvejetainę paiešką, kurios sudėtingumas yra $O(\log n)$.
   b) Pagal uždavinio sąlygą akivaizdu, kad labiausiai nevykusios strategijos sudėtingumas negali būti blogesnis už $O(n)$, nes pačiu nepalankiausiu atveju galima skaičių spėti iš viso n kartų. Tokia strategija galėtų būti iš eilės einančių skaičių spėjimas didėjimo ar mažėjimo tvarka arba tiesiog atsitiktinių skaičių spėjimas neatsižvelgiant į užuominas „skaičius didesnis“ arba „skaičius mažesnis“.
4. $11_2 + 11_3 + \cdots + 11_m = 2^1 + 2^0 + 3^1 + 3^0 + \cdots + m^1 + m^0 = 3 + 4 + \cdots + m + 1 =$
$$= \frac{3 + m + 1}{2} \cdot (m + 1 - 3 + 1) = \frac{(m + 4)(m - 1)}{2}.$$
   Jei darbą rašė VARDENIS PAVARDENIS, tai $m = 8^2 + 10^2 = 164$.
$$\frac{(164 + 4)(164 - 1)}{2} = 13692_{10} = 357(12)_{16} = \mathbf{357C_{16}}:$$
$$13692 = 855 \cdot 16 + 12,$$
$$855 = 53 \cdot 16 + 7,$$
$$53 = 3 \cdot 16 + 5,$$
$$3 = 0 \cdot 16 + 3.$$
5. a) Prisiminkime, kad Priuferio kode yra įrašomi vidinių medžio viršūnių numeriai, o pats medis turi dviem viršūnėm daugiau, nei Priuferio kode yra skaičių. Tokiu atveju medžio lapų skaičius yra lygus $2m + 2 - m = m + 2$. Kai $m = 164$, tai $m + 2 = 164 + 2 + 166$.
   b) Pavaizduokime pirmuosius medžius, kai $m = 1, 2, 3, \ldots$



   Įžvelgus dėsningumą, medžio centrą C galima apskaičiuoti bendruoju atveju pagal formulę
$$C = \begin{cases} \left\{\frac{m+1}{2}\right\}, \text{kai } m - \text{nelyginis skaičius,} \\ \left\{\frac{m}{2}, \frac{m}{2} + 1\right\}, \text{kai } m - \text{lyginis skaičius.} \end{cases}$$
   Kai $m = 164$, tai $C = \{82, 83\}$.

6. Žemiau pateiktas reiškinys teisingas su bet kuria $m \neq 0$ reikšme:

$$m \cdot (1 - 1/m) + 1 = m.$$

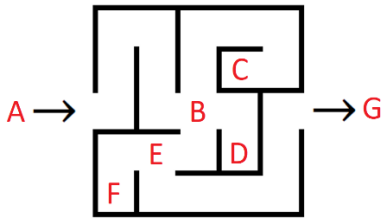
Taikant dėklo duomenų struktūrą šio reiškinio reikšmė galėtų būti apskaičiuojama taip:

stack_push(1)
stack_push(m)
stack_push(m)
stack_push(1)
stack_push(stack_pop() / stack_pop())
stack_push(1)
stack_push(stack_pop() − stack_pop())
stack_push(stack_pop() × stack_pop())
stack_push(stack_pop() + stack_pop())

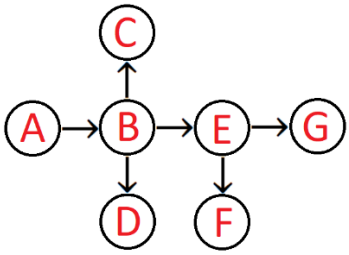
Tarkime, kad $m = 164$. Dėklo turinys kistų taip:

			1		1				
		164	164	$\frac{1}{164}$	$\frac{1}{164}$	$\frac{163}{164}$			
	164	164	164	164	164	164	163		
1	1	1	1	1	1	1	1	164	NULL

7. Pažymėkime labirinte pradžią ir pabaigą, kelio pasirinkimo vietas bei aklavietes kaip digrafo viršūnes:



Šį labirintą atitinka digrafas:



Vykdant paiešką į gylį šiame digrafe iš A viršūnės, laiko ir tėvystės atributai būtų apskaičiuojami šia tvarka:

π[A]=NULL, d[A]=1, π[B]=A, d[B]=2, π[C]=B, d[C]=3, f[C]=4, π[D]=B, d[D]=5, f[D]=6, π[E]=B, d[E]=7, π[F]=E, d[F]=8, f[F]=9, π[G]=E, d[G]=10 (labirintas pereitas).

Tęsiant paiešką toliau, likę atributai būtų apskaičiuojami taip: f[G]=11, f[E]=12, f[B]=13, f[A]=14.