Kelionės

LMIO 2017

3.2 etapas

Autorius: Justas Klimavičius

Sąlyga

- Baitlandijoje yra N miestų sunumeruotų nuo 1 iki N.
- Kiekvieną miestą j galima pasiekti traukiniu bėgiais iš miesto i tada ir tik tada, kai $i < j$.
- Rail Baitlandija siūlo bilietus tarp visų miestų išskyrus M porų.
- Martynas mėgsta keliauti.
- Kiekvieno miesto aplankymas Martynui suteikia kažkiek malonumo. Konkrečiau, i-tojo miesto aplankymas Martynui suteikia m_i malonumo.

Sąlyga

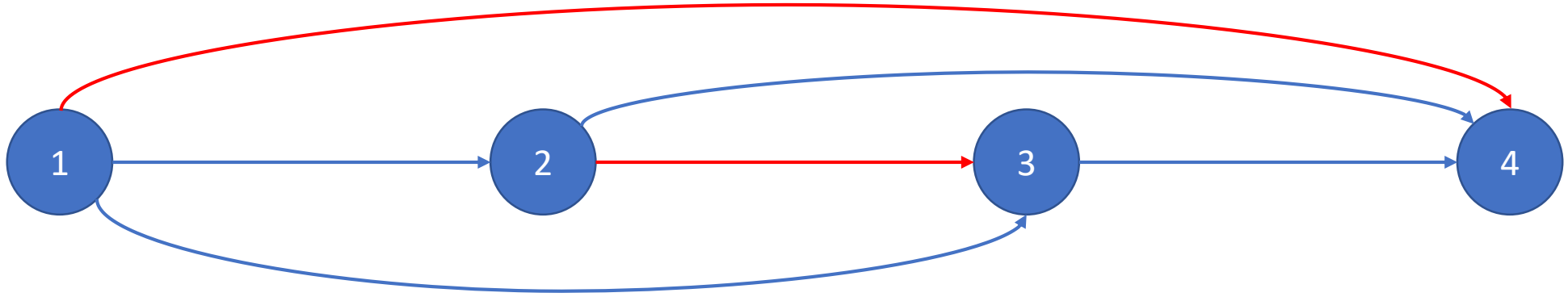
- Šiandien Martynas nori išbandyti visas įmanomas keliones, kurias siūlo *Rail Baitlandija*.
- T.y. tokia kelionė yra seka miestų $(a_1, a_2, \dots, a_k)$, tokių, kad visoms poroms (ai, ai_{+1}) *Rail Baitlandija* parduoda bilietus.
- Kelionė per miestus $(a_1, a_2, \dots, a_k)$ suteikia $m_{a1} + ma_2 + \dots + m_{ak}$. Skirtingų kelionių malonumas sumuojasi.
- Kiek malonumo iš viso patirs Martynas išbandęs visas tokias keliones?

Sąlyga kitaip

- Išimta M briaunų iš kryptinio, N viršūnių turinčio grafo, kuriame briauna (i, j) yra tada ir tik tada, kai $i < j$.

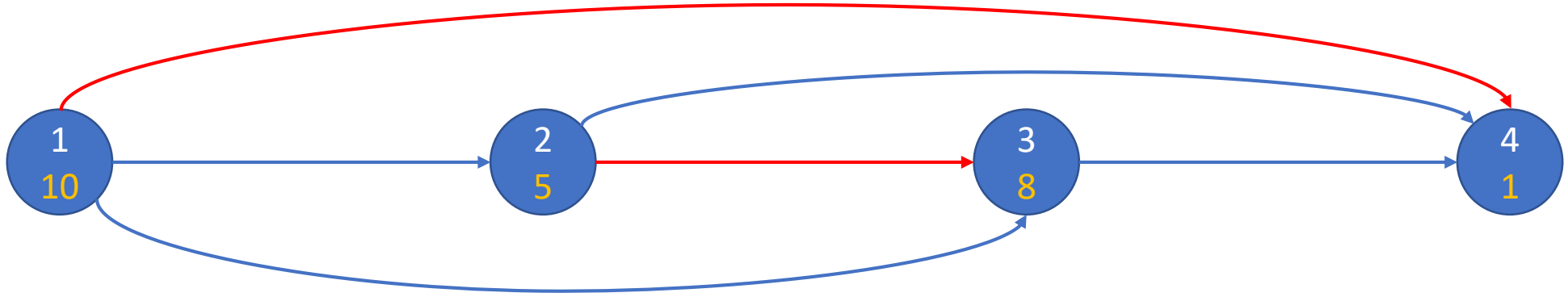
Sąlyga kitaip

- Išimta M briaunų iš kryptinio, N viršūnių turinčio grafo, kuriame briauna (i, j) yra tada ir tik tada, kai $i < j$.



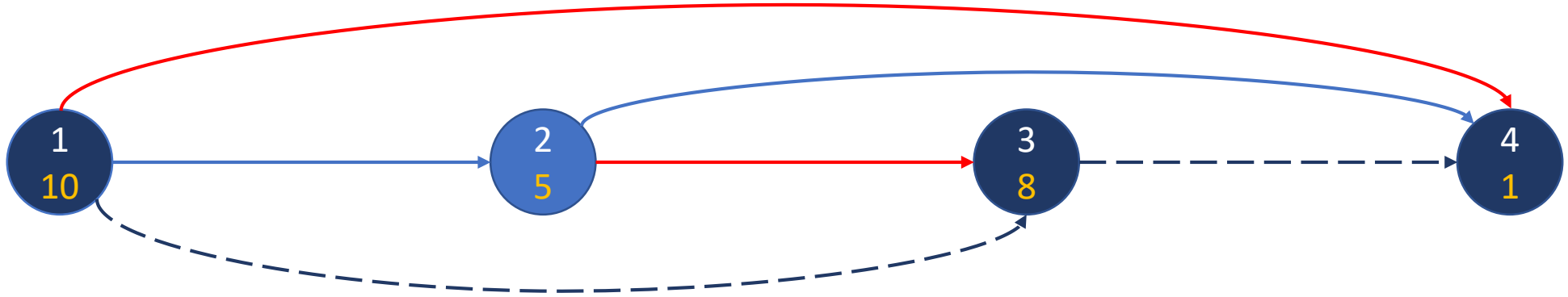
Sąlyga kitaip

- Išimta M briaunų iš kryptinio, N viršūnių turinčio grafo, kuriame briauna (i, j) yra tada ir tik tada, kai $i < j$.
- Kiekvienai grafo viršūnei i priskirtas svoris s_i.



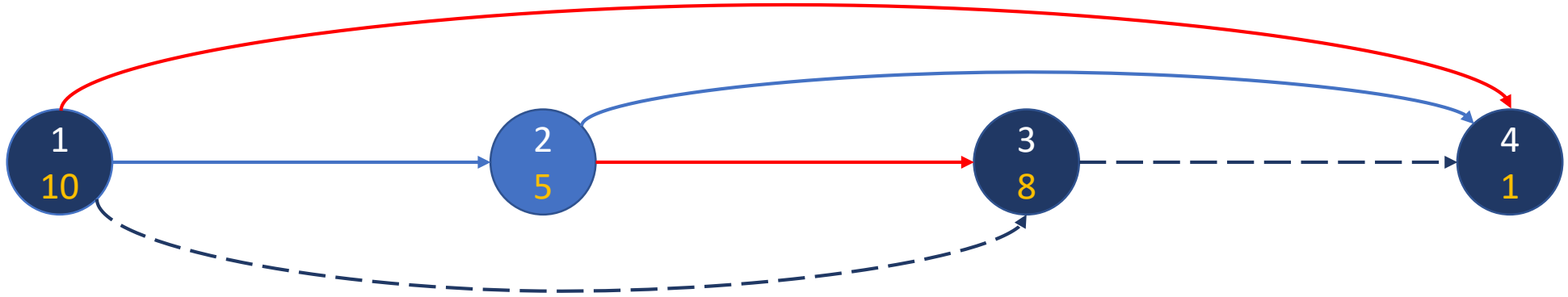
Sąlyga kitaip

- Kelias grafe – viršūnių seka $(a_1, \dots, a_k)$, $k > 1$, tokia kad grafe yra briauna (a_i, a_{i+1}), kai $1 \leq i < k$. (Paprasčiausiai kelias grafe...)



Sąlyga kitaip

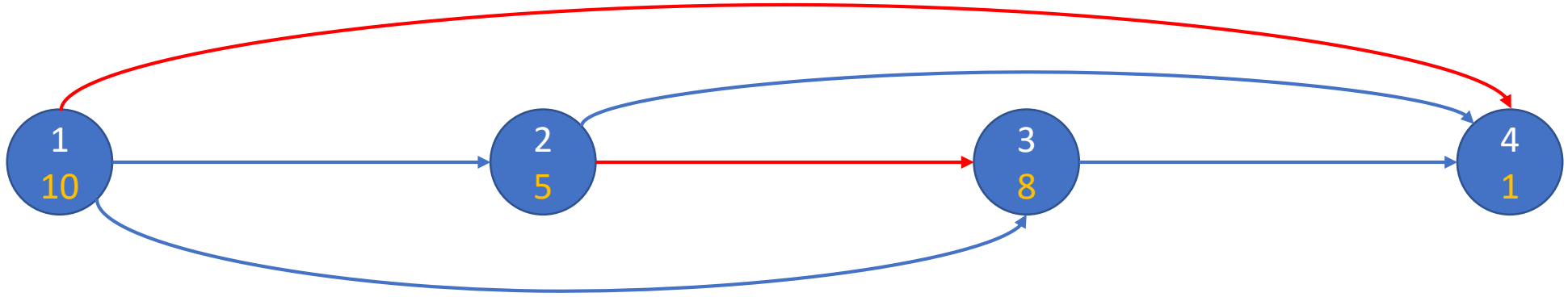
- Kelias grafe – viršūnių seka $(a_1, \dots, a_k)$, $k \geq 1$, tokia kad grafe yra briauna (a_i, a_{i+1}), kai $1 \leq i < k$. (Paprasčiausiai kelias grafe...)
- Kelio $(a_1, \dots, a_k)$ svoris – $\sum_{i=1}^{k} s_{a_i}$.



Šiuo atveju svoris: $10 + 8 + 1 = 19$

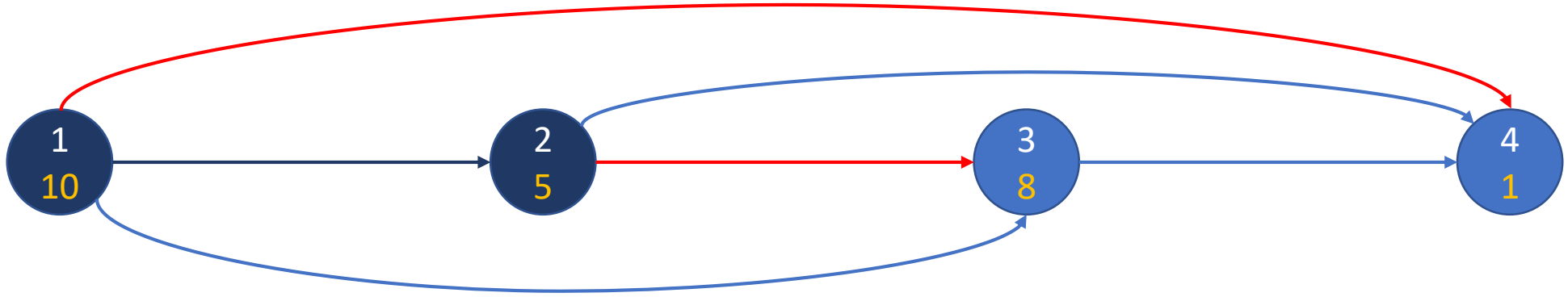
Sąlyga kitaip

- Užduotis – rasti visų grafe esančių kelių svorį.



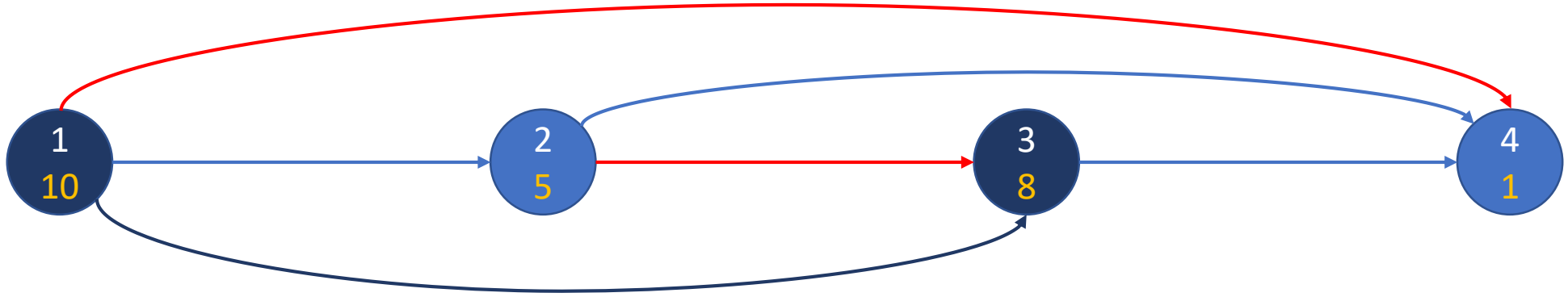
Sąlyga kitaip

- Užduotis – rasti visų grafe esančių kelių svorį.
- $(10 + 5)$



Sąlyga kitaip

- Užduotis – rasti visų grafe esančių kelių svorį.
- $(10 + 5) + (10 + 8)$

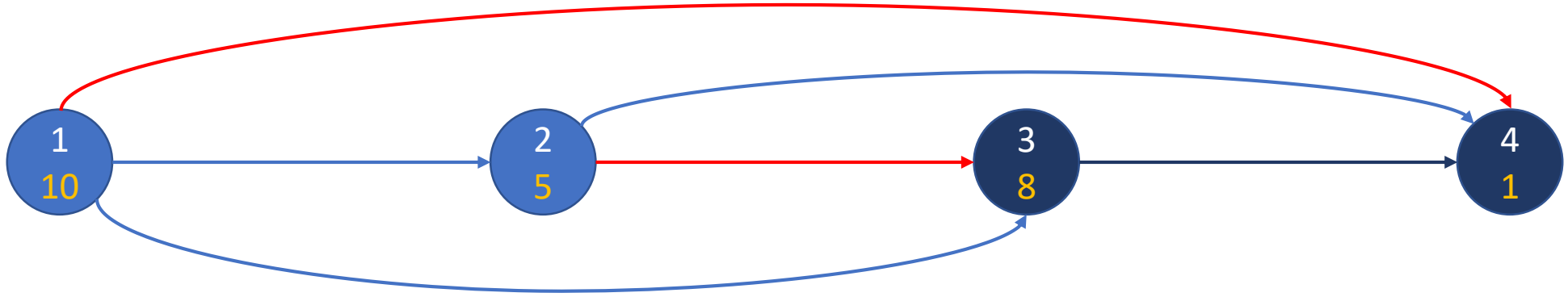


Sąlyga kitaip

- Užduotis – rasti visų grafe esančių kelių svorį.
- $(10 + 5) + (10 + 8) + (5 + 1)$

Sąlyga kitaip

- Užduotis – rasti visų grafe esančių kelių svorį.
- $(10 + 5) + (10 + 8) + (5 + 1) + (8 + 1)$

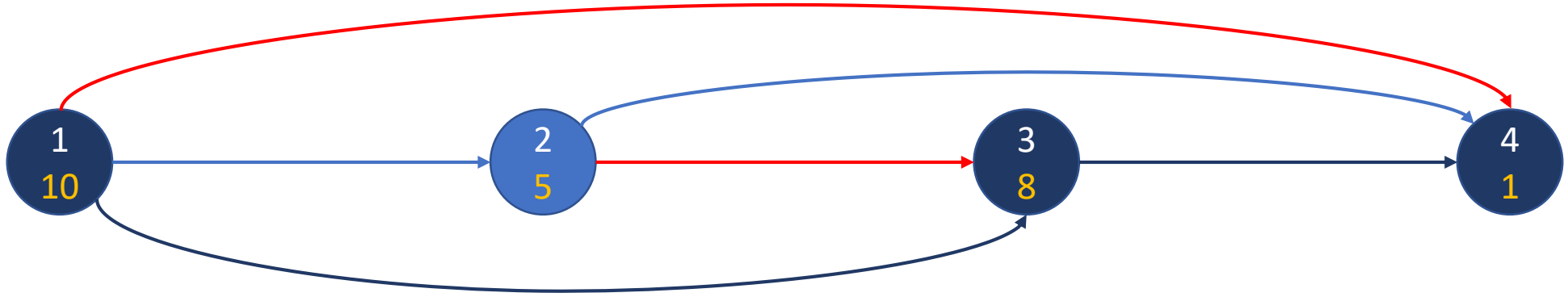


Sąlyga kitaip

- Užduotis – rasti visų grafe esančių kelių svorį.
- $(10 + 5) + (10 + 8) + (5 + 1) + (8 + 1) + (10 + 5 + 1)$

Sąlyga kitaip

- Užduotis – rasti visų grafe esančių kelių svorį.
- $(10 + 5) + (10 + 8) + (5 + 1) + (8 + 1) + (10 + 5 + 1) + (10 + 8 + 1) = 83$



Dalinės užduotys

- $N \leq 20$
- $N \leq 500$
- $N \leq 5000$
- Visose dalinėse užduotyse $N, M \leq 200000$

Pirma dalinė užduotis. 8t.

Pirma dalinė užduotis. 8t.

- Iš kiekvienos viršūnės paleidžiame paiešką platyn.
- Atėję į jau aplankytą viršūnę nenutraukiame paieškos. Taip išnaršysime visus kelius.
- Saugome kol kas nueito kelio svorį ir kiekvieną kartą pridedam jį prie bendro atsakymo.

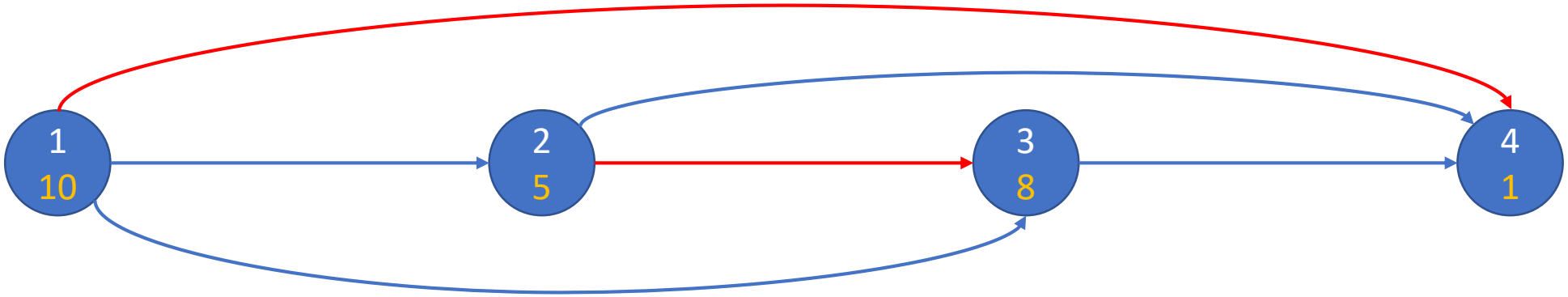
Antra dalinė užduotis. 14t.

Antra dalinė užduotis. 14t.

Kokia idėja??

Dinaminis Programavimas

Antra dalinė užduotis. 14t.



A[l, i] – kiek yra kelių, pasibaigiančių viršūnėje i, kurių ilgis yra l?

B[l, i] – kokia yra svorių suma visų šitų kelių?

Jei i turi kaimynes $x_1, \ldots, x_k$ iš kairės, tai

$A[l,i] = A[l-1, x_1] + A[l-1, x_2] + \ldots + A[l-1, x_k]$ O(N^3)

$B[l,i] = B[l-1,x_1] + A[l-1,x_1] * s[i] + \ldots + B[l-1,x_k] + A[l-1,x_k] * s[i] =$

$= (B[l-1,x_1] + \ldots + B[l-1,x_k]) + s[i] * (A[l-1,x_1] + \ldots + A[l-1, x_k])$ O(N^3)

Trečia dalinė užduotis. 25t.

Trečia dalinė užduotis. 25t.

A[i] – kelių, pasibaigiančių viršūnėje i, skaičius.

B[i] – kelių, pasibaigiančių viršūnėje i, svorių suma.

Jei viršūnė i turi kaimynes x_1, .., x_k, tai

$A[i] = A[x_1] + .. + A[x_k] + k$ O(N^2)

$B[i] = B[x_1] + A[x_1] * s[i] + \ldots + B[x_k] + A[x_k] * s[i] + (s[x_1] + s[i]) + \ldots + (s[x_k] + s[i]) =$

$B[x_1] + .. + B[x_k] + s[i] * (A[x_1] + \ldots + A[x_k]) + (s[x_1] + \ldots + s[x_k]) + k * s[i]$

O(N^2)

Ketvirta dalinė užduotis. 53t.

Ketvirta dalinė užduotis. 53t.

A[i] – kelių, pasibaigiančių viršūnėje i, skaičius.

B[i] – kelių, prasidedančių viršūnėje i, skaičius.

$A[i] = A[x_1] + .. + A[x_k] + k$

Tarkime į viršūnę i neateina briaunos iš t_i viršūnių $y_1, \ldots, y_{ti}$.

Tada

$A[i] = A[1] + A[2] + \ldots + A[i-1] - A[y_1] - A[y_2] - \ldots - A[y_{ti}] + k$

Ketvirta dalinė užduotis. 53t.

$A[i] = A[1] + A[2] + \ldots + A[i-1] - A[y_1] - A[y_2] - \ldots - A[y_{ti}] + k$

Paskaičiuokime $P[i] = A[1] + \ldots + A[i]$.

$P[1] = A[1]$

$P[i] = P[i-1] + A[i]$ $\quad O(N)$

Tada

$A[i] = P[i-1] - A[y_1] - \ldots - A[y_{ti}] + k$

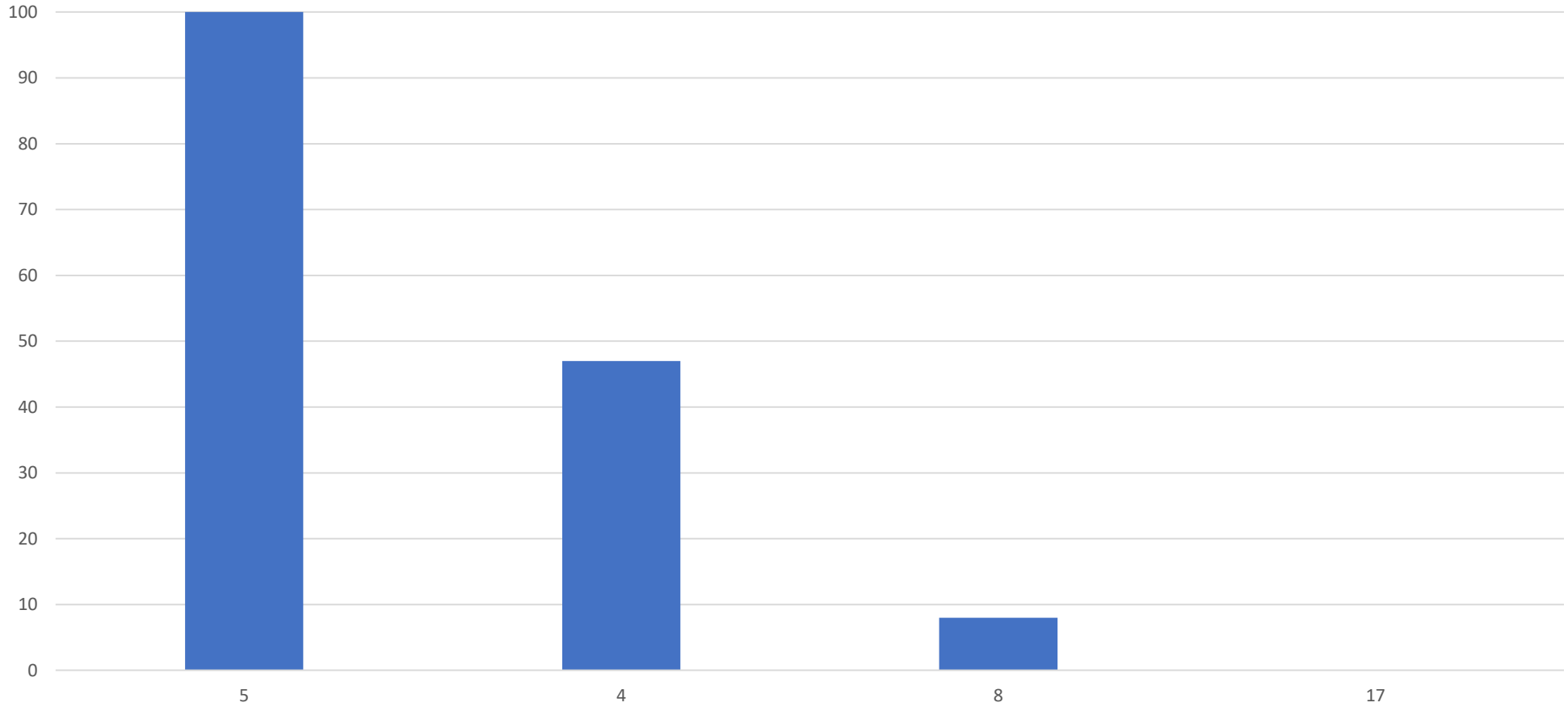
Apskaičiuojant visus A[i], iš viso reikės atimti $t_1 + t_2 + \ldots + t_N = M$ narių, todėl sudėtingumas yra $O(N + M)$.

Ketvirta dalinė užduotis. 53t.

A[i] – kelių, pasibaigiančių viršūnėje i, skaičius.
B[i] – kelių, pasibaigiančių viršūnėje i, svorių suma.

Analogiškai galime apskaičiuoti masyvą B[i].

Iš viso kelių einančių per viršūnę i – (A[i] + 1) * (B[i] + 1) – 1

Atsakymas:
s[1] * ((A[1] + 1) * (B[1] + 1) - 1) + … + s[N] * ((A[N] + 1) * (B[N] + 1) - 1)

Statistika