

# Diskrečioji matematika, IT 1 kursas

Dr. Robertas Petuchovas

Vilniaus universitetas

2016 - 2017 m.

# V skyrius.

## Algoritmų analizė

### 5.2 Sumavimas ir uždaros formos

**Ap.** Matematinį reiškinį vadiname uždara forma, jeigu jis gali būti apskaičiuotas su jo argumentais atliekant baigtinį veiksmų skaičių.

*Pavyzdžiui.* Reiškiny

$$2 + 4 + \dots + 2n$$

nėra uždara forma, bet

$$n(n + 1)$$

yra uždara forma.

## Sumavimo žymuo

$$\sum_{k=1}^n a_k = a_1 + \dots + a_n.$$

Kai kurios sumavimo savybės:

1.

$$\sum ca_k = c \sum a_k,$$

2.

$$\sum (a_k + b_k) = \sum a_k + \sum b_k,$$

3.

$$\sum a_k x^{i+k} = x^i \sum a_k x^k,$$

4.

$$\sum_{k=m}^n a_{k+i} = \sum_{k=m+i}^{n+i} a_k,$$

5.

$$\sum_{k=1}^n (a_k - a_{k-1}) = a_n - a_0$$

ir

$$\sum_{k=1}^n (a_{k-1} - a_k) = a_0 - a_n.$$

## Kai kurios naudingos uždaros formos:

1.

$$\sum_{k=m}^n c = (n - m + 1)c.$$

2.

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}.$$

3.

$$\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}.$$

4.

$$\sum_{k=0}^n a^k = \frac{a^{n+1} - 1}{a - 1},$$

kai  $a \neq 1$ .

5.

$$\sum_{k=1}^n ka^k = \frac{a - (n+1)a^{n+1} + na^{n+2}}{(a-1)^2},$$

kai  $a \neq 1$ .

1 pavyzdys. Rasime sumos  $\sum_{k=2}^n (k-1)2^{k+1}$  uždara formą.

Sprendimas:

$$\sum_{k=2}^n (k-1)2^{k+1} = \sum_{k=1}^{n-1} k2^{k+2} \quad (4 \text{ savybė})$$

$$= 2^2 \sum_{k=1}^{n-1} k2^k \quad (3 \text{ savybė})$$

$$= 2^2(2 - n2^n + (n-1)2^{n+1}) \quad (5 \text{ forma})$$

$$= 2^3 + (n-2)2^{n+2}.$$

2 pavyzdys. Rasime sumos

$$2 + 2^2 \cdot 7 + 2^3 \cdot 14 + \dots + 2^n \cdot (n-1) \cdot 7$$

uždara formą.

*Sprendimas:*

$$2 + \sum_{k=2}^n 2^k (k-1)7 = 2 + 7 \sum_{k=2}^n (k-1)2^k \quad (1 \text{ savybė})$$

$$= 2 + 7 \sum_{k=1}^{n-1} k2^{k+1} \quad (4 \text{ savybė})$$

$$= 2 + 14 \sum_{k=1}^{n-1} k2^k \quad (3 \text{ savybė})$$

$$= 2 + 14(2 - n2^n + (n-1)2^{n-1}). \quad (5 \text{ forma})$$

1 užduotis. Pritaikę sumavimo savybes, įrodykite, kad

$$2 + 3 + \dots + n = \frac{(n-1)(n+2)}{2}.$$

2 užduotis. Pritaikę sumavimo savybes, raskite sumos

$$3 + 7 + \dots + (3 + 4n)$$

uždara formą.

Atsakymas:  $(2n + 3)(n + 1)$ .



3 pavyzdys. Tegul  $\text{count}(n)$  yra žemiau esančiame algoritme atliekamų priskyrimų ( $:=$ ) skaičius.

$i := 1;$  (1)

**while**  $i < n$  **do**

$i := i + 1;$  ( $n - 1$ )

**for**  $j := 1$  **to**  $i$  **do**  $S$  **od** ( $2 + 3 + \dots + n$ )

**od,**

čia  $n \in \mathbb{N}$ . Reiškiniai tarp skliaustų parodo kiek kartų buvo įvykdytas priskyrimas  $:=$ . Taigi,

$$\begin{aligned}\text{count}(n) &= 1 + (n - 1) + (2 + 3 + \dots + n) \\ &= (n - 1) + (1 + 2 + \dots + n) \\ &= n - 1 + \frac{n(n + 1)}{2}.\end{aligned}$$

*Užduotis.* Tegul  $\text{count}(n)$  yra veiksmo  $S$  įvykdymų aukščiau nurodytame algoritme skaičius. Raskite funkcijos  $\text{count}(n)$  uždara formą.

*Sprendimas:*

$$\begin{aligned}\text{count}(n) &= (2 + 3 + \dots + n) \\ &= (1 + 2 + 3 + \dots + n) - 1 \\ &= \frac{n(n+2)}{2} - 1.\end{aligned}$$

4 pavyzdys. Tegul  $\text{count}(n)$  yra  $S$  veiksmo atlikimų skaičius pateiktame algoritme:

$i := 1;$

**while**  $i < n$  **do**

$i := i + 2;$

**for**  $j := 1$  **to**  $i$  **do**  $S$  **od**

**od,**

kuriame  $n \in \mathbb{N}$ . Rasime funkcijos  $\text{count}(n)$  uždara formą.

*Sprendimas:* pastebėkime, kad parametro  $i$  reikšmės prieš kiekvieną for ciklą yra tokios:

$$3, 5, \dots, (2k + 1);$$

čia  $i = 2k + 1 \geq n$  yra while ciklo sustojimo taškas.

Gauname

$$\begin{aligned}\text{count}(n) &= 3 + 5 + \dots + (2k + 1) \\ &= \sum_{i=1}^k (2i + 1) = 2 \sum_{i=1}^k i + \sum_{i=1}^k 1 \\ &= \frac{2k(k + 1)}{2} + k = k(k + 2),\end{aligned}$$

bet mums reikia  $\text{count}(n)$  išreikšti per  $n$ . Kadangi  $2k + 1 \geq n$  yra while ciklo sustojimo taškas, tai  $2k - 1 < n$  yra taškas, su kuriuo while sąlyga buvo paskutinį kartą išpildyta, todėl

$$2k - 1 < n \leq 2k + 1.$$

Išsprendę pastarąsias nelygybes  $k$  atžvilgiu, gauname  $k = \lceil (n - 1)/2 \rceil$ , todėl

$$\text{count}(n) = k(k + 2) = \lceil (n - 1)/2 \rceil^2 + 2\lceil (n - 1)/2 \rceil.$$

# Rekomenduojamos literatūros sąrašas

-  James L. Hein, "Discrete Structures, Logic, and Computability"
-  + StudentStudyGuide.pdf