KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Modulio P170B400 „Algoritmų sudarymas ir analizė“

Laboratorinio darbo ataskaita

Pirmas laboratorinis darbas

Dėstytojas
Lekt. Makackas Dalius

Studentas
Rokas Puzonas

KAUNAS, 2023

Turinys

1. PIRMA UŽDUOTIS

1. Sąlyga

- Realizuoti metodą, kuris atitiktų pateiktos rekurentinės lygties sudėtingumą, t. y. programinio kodo rekursinių iškvietimų ir kiekvieno iškvietimo metu atliekamų veiksmų priklausomybę nuo duomenų. Metodas per parametrus turi priimti masyvą, kurio duomenų kiekis yra rekurentinės lygties kintamasis n (arba masyvą ir indeksų rėžius, kurie atitinkamai nurodo masyvo nagrinėjamų elementų indeksus atitinkamame iškvietime)
- Kiekvienam realizuotam metodui atlikti programinio kodo analizę, parodant jog jis atitinka pateiktą rekurentinę lygtį.
- Išspręskite rekurentinę lygtį ir apskaičiuokite jos asimptotinį sudėtingumą (taikoma pagrindinė teorema, medžių ar kitas sprendimo metodas).
- Atlikti eksperimentinį tyrimą (našumo testus) ir patikrinkite ar apskaičiuotas metodo asimptotinis sudėtingumas atitinka eksperimentinius rezultatus.

Lygtis: $T(n) = 2 * T\left(\frac{n}{9}\right) + n^5$

2. Kodo analizavimas

```
public static int T1(int[] n, int size)
{
    if (size < 1)
        return 0;

    int sum = 0;
    for (int i = 0; i < Math.Pow(size, 5); i++)
    {
        sum++;
    }
    return T1(n, size / 9) + T1(n, size / 9) + sum;
}
```

Laikas	Kiekis
c1	1
c2	1
c3	1
c4	n^5+1
c5	n^5
$2T\left(\frac{n}{9}\right) + c6$	1

$$T(n) = \begin{cases} c1 + c2, n < 1 \\ c1 + c3 + c4 * (n^5 + 1) + c5 * n^5 + 2T\left(\frac{n}{9}\right) + c6 \end{cases}$$

$$T(n) = 2T\left(\frac{n}{9}\right) + n^5 * (c5 + c4) + c1 + c2 + c3 + c4 + c6$$

3. Lygties sprendimas

$$T(n) = 2T\left(\frac{n}{9}\right) + n^5$$

$$a = 2 \quad b = 9 \quad f(n) = n^5$$

$$n^{\log_b a} = n^{\log_9 2} = n^{0.31} \quad 3 \text{ atvejis}$$

$$F(n) = \Omega\left(n^{\log_9 2 + \varepsilon}\right)$$

$$\varepsilon < 5 - 0{,}31 \Rightarrow \varepsilon = 4$$

$$\lim_{n\to\infty} \frac{(n^5)'}{(n^{4,31})'} = \lim_{n\to\infty} \frac{(5n^4)'}{(4{,}31 n^{3,31})'} = \lim_{n\to\infty} \frac{(20n^3)'}{(14.2661 n^{2,31})'}$$

$$= \lim_{n\to\infty} \frac{(60n^2)'}{(32{,}954691 n^{1,31})'}$$

$$= \lim_{n\to\infty} \frac{(60n^2)'}{(32{,}954691 n^{1,31})'}$$

$$= \lim_{n\to\infty} \frac{(120n)'}{(43.17064521 n^{0,31})'}$$

$$= \lim_{n\to\infty} \frac{120 n^{0,69}}{13{,}3829000151}$$

$$= \infty$$

$$a F\left(\frac{n}{b}\right) \leqslant c F(n)$$

$$2 \cdot \left(\frac{n}{9}\right)^5 \leqslant c n^5$$

$$\frac{2n^5}{59049} \leqslant c n^5$$

$$C = \frac{1}{2} \Rightarrow \frac{2n^5}{59049} \leqslant \frac{n^5}{2} \Rightarrow T(n) = \Theta(n^5)$$

4. Kodo našumo testavimas

Našumo testavimo rezultatai atitinka n^5 asimptotinį sudėtingumą. Analitiniai ir eksperimentiniai rezultatai sutampa.

2. ANTRA UŽDUOTIS

1. Sąlyga

- Realizuoti metodą, kuris atitiktų pateiktos rekurentinės lygties sudėtingumą, t. y. programinio kodo rekursinių iškvietimų ir kiekvieno iškvietimo metu atliekamų veiksmų priklausomybę nuo duomenų. Metodas per parametrus turi priimti masyvą, kurio duomenų kiekis yra rekurentinės lygties kintamasis n (arba masyvą ir indeksų rėžius, kurie atitinkamai nurodo masyvo nagrinėjamų elementų indeksus atitinkamame iškvietime)
- Kiekvienam realizuotam metodui atlikti programinio kodo analizę, parodant jog jis atitinka pateiktą rekurentinę lygtį.
- Išspręskite rekurentinę lygtį ir apskaičiuokite jos asimptotinį sudėtingumą (taikoma pagrindinė teorema, medžių ar kitas sprendimo metodas).
- Atlikti eksperimentinį tyrimą (našumo testus) ir patikrinkite ar apskaičiuotas metodo asimptotinis sudėtingumas atitinka eksperimentinius rezultatus.

Lygtis: $T(n) = T\left(\frac{n}{6}\right) + T\left(\frac{n}{7}\right) + n$

2. Kodo analizavimas

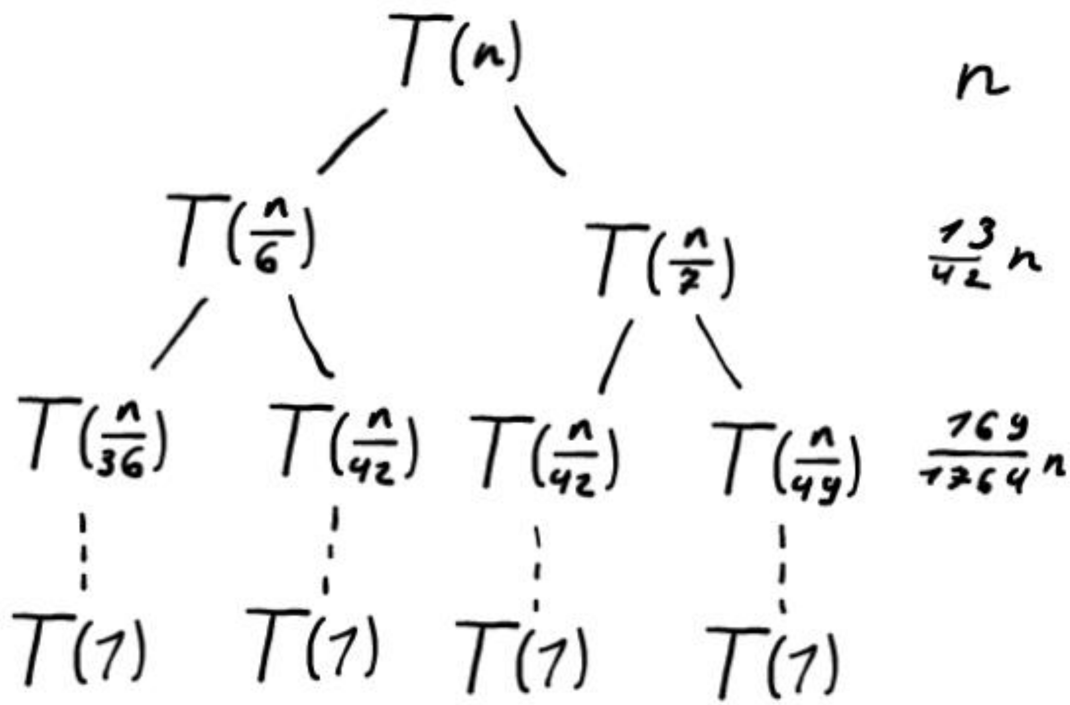
Kodas	Laikas	Kiekis
`public static int T2(int[] n, int size)`		
`{`		
`    if (size < 1)`	c1	1
`        return 0;`	c2	1
`    int sum = 0;`	c3	1
`    for (int i = 0; i < size; i++)`	c4	n+1
`    {`		
`        sum++;`	c5	n
`    }`		
`    return T2(n, size / 6) + T2(n, size / 7) + sum;`	$T\left(\frac{n}{6}\right) + T\left(\frac{n}{7}\right) + c6$	1
`}`		

$$T(n) = \begin{cases} c1 + c2, n < 1 \\ c1 + c3 + c4 * (n+1) + c5 * n + T\left(\frac{n}{6}\right) + T\left(\frac{n}{7}\right) + c6 \end{cases}$$

$$T(n) = T\left(\frac{n}{6}\right) + T\left(\frac{n}{7}\right) + n * (c5 + c4) + c1 + c2 + c3 + \ c4 + c6$$

3. Lygties sprendimas

$$T(n)=T\left(\frac{n}{6}\right)+T\left(\frac{n}{7}\right)+n$$



$$\left(\frac{1}{6}+\frac{1}{7}\right)^i=\left(\frac{13}{42}\right)^i$$

Ilgiausia šaka: $n+\frac{n}{6}+\frac{n}{36}+\ldots+1$

Trumpiausia šaka: $n+\frac{n}{7}+\frac{n}{49}+\ldots+1$

Aukštis : $\log_7 n \leq h \leq \log_6 n$

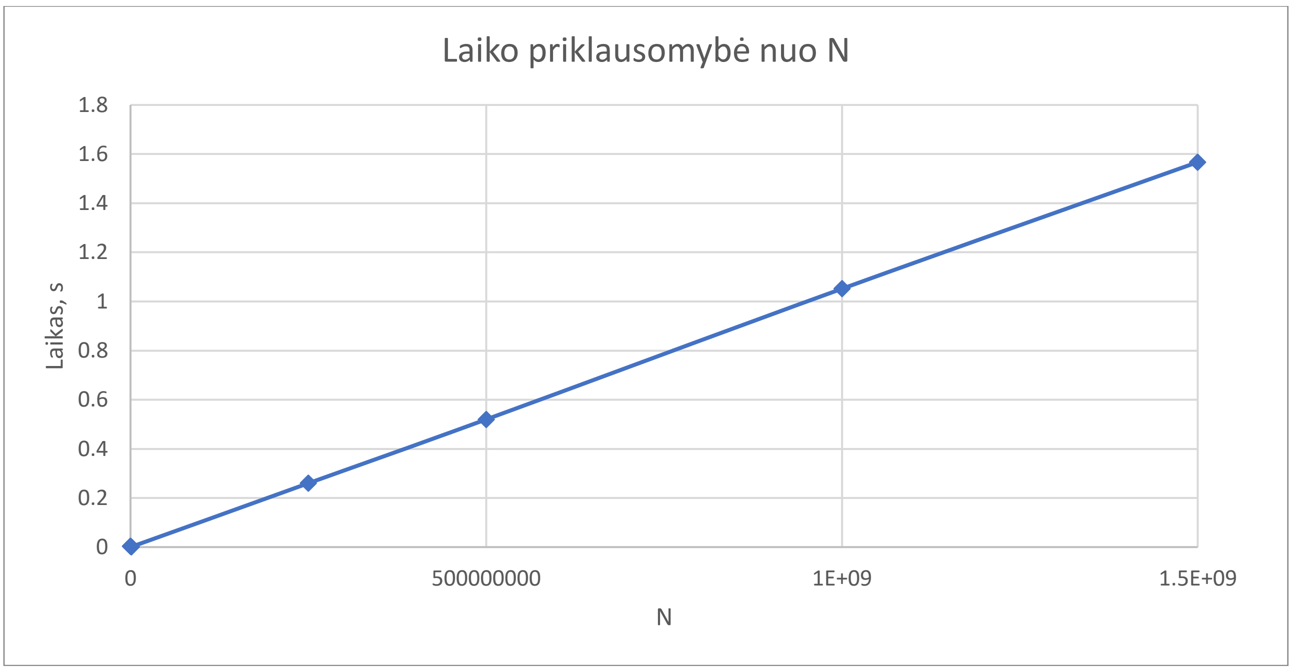
$$T(n)=n\sum_{i=0}^{h}\left(\frac{13}{42}\right)^i < n\sum_{i=0}^{\infty}\left(\frac{13}{42}\right)^i=\frac{n}{1-\frac{13}{42}}=\frac{42}{29}n \qquad T(n)=O(n)$$

$$T(n) = n\sum_{i=0}^{n}\left(\frac{13}{42}\right)^{i} \geq n\sum_{i=0}^{\lfloor \log_3 n \rfloor}\left(\frac{13}{42}\right)^{i} = \frac{n\left(\frac{13}{42}\right)^{\lfloor \log_3 n \rfloor + 1}}{\frac{13}{42} - 1}$$

$$= \frac{29}{42} n \left(1 - \left(\frac{13}{42}\right)^{\lfloor \log_3 n \rfloor + 1}\right) \geq n \qquad T(n) = \Omega(n)$$

$T(n) = \Theta(n)$, nes $n \leq T(n) \leq \frac{42}{29} n$ visiem $n > 0$

4. Kodo našumo testavimas



Grafikas atitinka analitinius rezultatus kuris yra tiesinis sudėtingumas.

3. TREČIA UŽDUOTIS

1. Sąlyga

- Realizuoti metodą, kuris atitiktų pateiktos rekurentinės lygties sudėtingumą, t. y. programinio kodo rekursinių iškvietimų ir kiekvieno iškvietimo metu atliekamų veiksmų priklausomybę nuo duomenų. Metodas per parametrus turi priimti masyvą, kurio duomenų kiekis yra rekurentinės lygties kintamasis n (arba masyvą ir indeksų rėžius, kurie atitinkamai nurodo masyvo nagrinėjamų elementų indeksus atitinkamame iškvietime)
- Kiekvienam realizuotam metodui atlikti programinio kodo analizę, parodant jog jis atitinka pateiktą rekurentinę lygtį.
- Išspręskite rekurentinę lygtį ir apskaičiuokite jos asimptotinį sudėtingumą (taikoma pagrindinė teorema, medžių ar kitas sprendimo metodas).
- Atlikti eksperimentinį tyrimą (našumo testus) ir patikrinkite ar apskaičiuotas metodo asimptotinis sudėtingumas atitinka eksperimentinius rezultatus.

Lygtis: $T(n) = T(n-8) + T(n-6) + n$

2. Kodo analizavimas

```
"
public static int T3(int[] n, int size)
{
    if (size < 1)
        return 0;

    int sum = 0;
    for (int i = 0; i < size; i++)
    {
        sum++;
    }
    return T3(n, size - 8) + T3(n, size - 6) + sum;
}
```

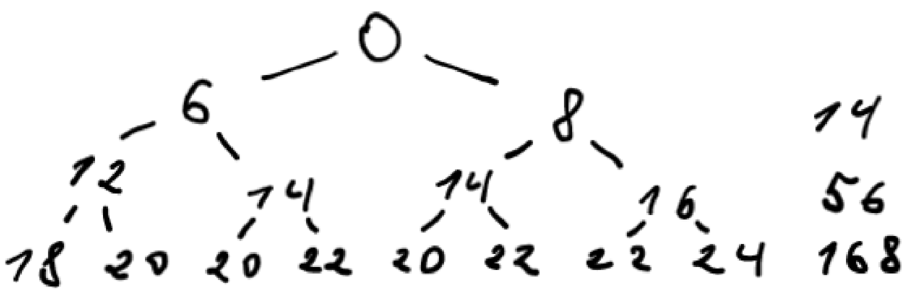
Eilutė	Laikas	Kiekis
`if (size < 1)`	c1	1
`return 0;`	c2	1
`int sum = 0;`	c3	1
`for (int i = 0; i < size; i++)`	c4	n+1
`sum++;`	c5	n
`return T3(n, size - 8) + T3(n, size - 6) + sum;`	$T(n-8) + T(n-6) + c6$	1

$$T(n) = \begin{cases} c1 + c2, n < 1 \\ c1 + c3 + c4 * (n+1) + c5 * n + T(n-8) + T(n-6) + c6 \end{cases}$$

$$T(n) = T(n-8) + T(n-6) + n * (c4 + c5) + c1 + c2 + c3 + c4 + c6$$

3. Lygties sprendimas

$$T(n) = T(n-8) + T(n-6) + n$$



$$\begin{cases} S_0 = 0 \\ S_n = 2S_{n-1} + 14 \cdot 2^{n-1} \end{cases} \qquad S_n = 14n\,2^{n-1}$$

$$S_{n+1} = 2 \cdot 14\,n\,2^{n-1} + 14 \cdot 2^n$$

$$T(n) = \sum_{i=0}^{h} \left(2^i n - 14\, i\, 2^{i-1}\right) = n\sum_{i=0}^{h} 2^i - 14 \sum_{i=0}^{h} i\,2^{i-1} =$$

$$= n\left(2^{h+1} - 1\right) + 14\left(h2^h - 2^h + 1\right)$$

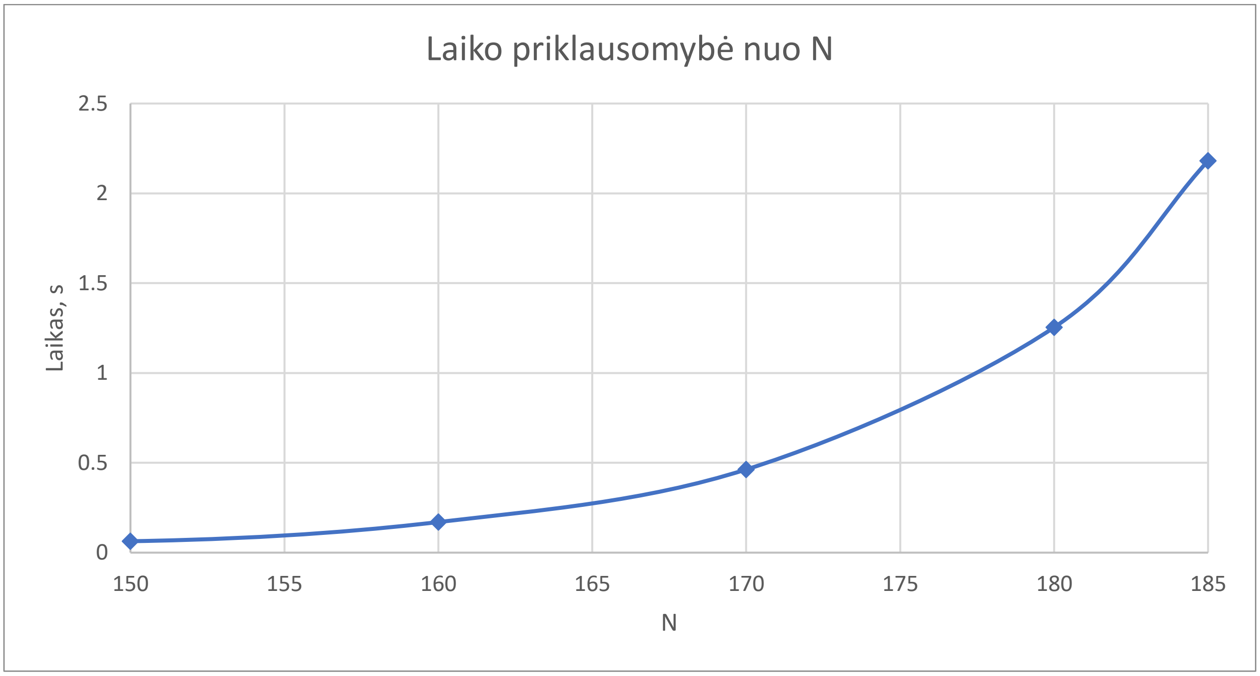
Apatinis vertinimas, kai $h = \lfloor \frac{n}{8} \rfloor$

$$T(n) = \Omega\left(n 2^{\frac{n}{8}}\right)$$

Viršutinis vertinimas, kai $h = \lfloor \frac{n}{6} \rfloor$

$$T(n) = O\left(n 2^{\frac{n}{6}}\right)$$

4. Kodo našumo testavimas

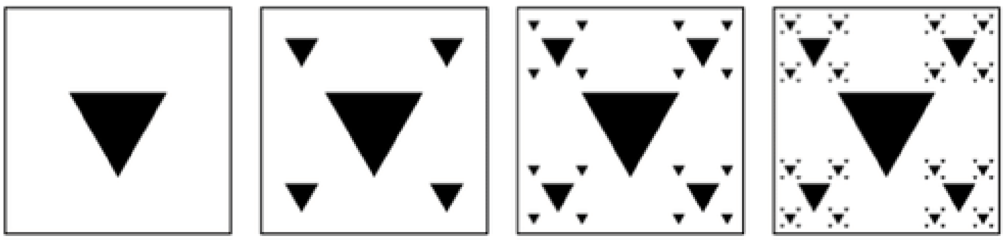


Grafikas atitinka 2^n kitimo greitį, todėl ir atitinka $O(n2^{\frac{n}{6}})$ anlitinius rezultatus.

4. KETVIRTA UŽDUOTIS

1. Sąlyga

- Programos rezultatas BMP formato bylos demonstruojančios programos rekursijas.
- Eksperimentiškai nustatykite darbo laiko ir veiksmų skaičiaus priklausomybę nuo generuojamo paveikslėlio dydžio (taškų skaičiaus). Gautus rezultatus atvaizduokite grafikais. Grafiką turi sudaryti nemažiau kaip 5 taškai ir paveikslėlio taškų skaičius turi didėti proporcingai (kartais).
- Analitiškai įvertinkite procedūros, kuri generuoja paveikslėlį, veiksmų skaičių sudarydami rekurentinę lygtį ir ją išspręskite. Gautas rezultatas turi patvirtinti eksperimentinius rezultatus.



2. Kodo analizavimas

```
public static void TrianglesRecursive(BMPImage image, int x, int y, uint width, uint height)
{
    if (width <= 3 || height <= 3)
        return;

    image.Triangle(
        x + (int)(width / 3 * 1  ), (y + (int)height / 3 * 2),
        x + (int)(width / 3 * 2  ), (y + (int)height / 3 * 2),
        x + (int)(width / 3 * 1.5), (y + (int)height / 3 * 1),
        new Color(0, 0, 0)
    );

    TrianglesRecursive(
        image, x, y,
        width/3, height/3
    );
    TrianglesRecursive(
        image, x + (int)width/3*2, y,
        width / 3, height / 3
    );
    TrianglesRecursive(
        image, x, y + (int)height/3*2,
        width / 3, height / 3
    );
    TrianglesRecursive(
        image, x + (int)width / 3 * 2, y + (int)height / 3 * 2,
        width / 3, height / 3
    );
}
```

Laikas	Kiekis
c1	1
c2	1
n^2	1
$T\left(\frac{n}{3}\right)$	4

$$T(n) = \begin{cases} c1 + c2, n \leq 3 \\ c1 + n^2 + 4T\left(\frac{n}{3}\right) \end{cases}$$

$$T(n) = 4T\left(\frac{n}{3}\right) + n^2 + c1 + c2$$

3. Lygties analizė

$$T(n) = 4T\left(\frac{n}{3}\right) + n^2$$

$a = 4 \quad b = 3 \quad f(n) = n^2$

$n^{\log_b a} = n^{\log_3 4} = n^{1,26}$ 3 atvejis

$F(n) = \Omega\left(n^{\log_3 4 + \varepsilon}\right)$

$\varepsilon < 2 - 1,26 \Rightarrow \varepsilon = 0,5$

$$\lim_{n \to \infty} \frac{(n^2)'}{(n^{1,5})'} = \lim_{n \to \infty} \frac{(2n)'}{(1,5n^{0,5})'} = \lim_{n \to \infty} \frac{2n^{0,5}}{0,75} = \infty$$

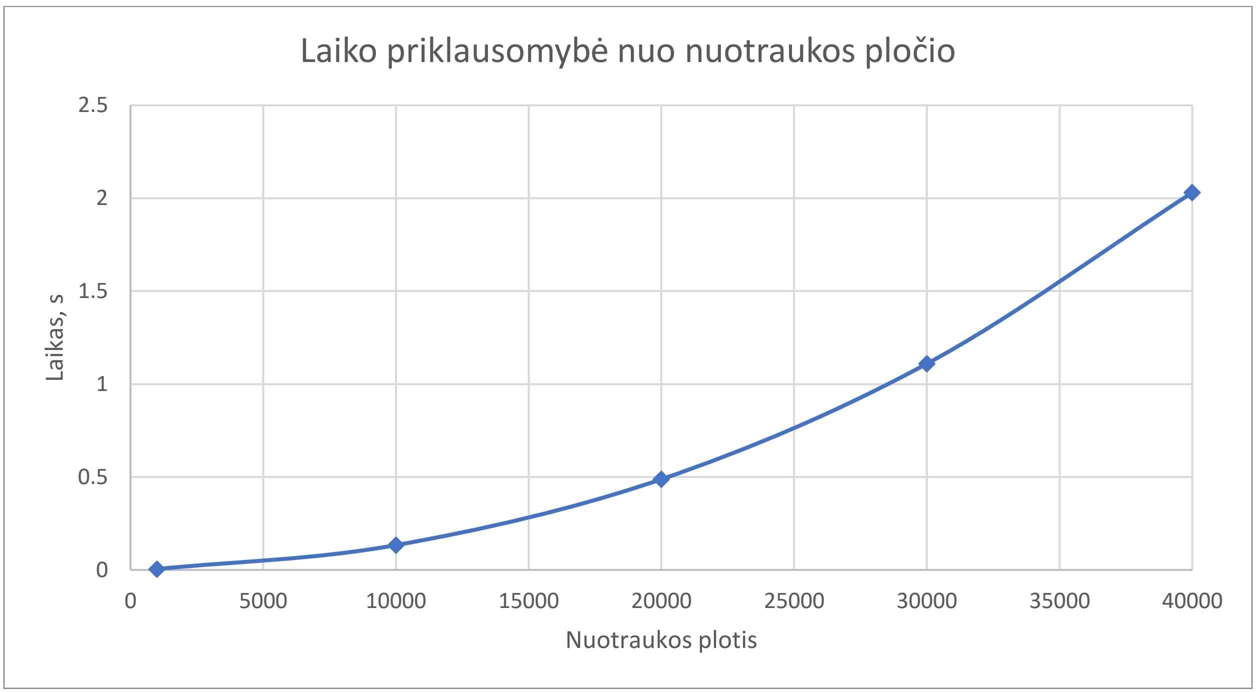
$$a F\left(\frac{n}{b}\right) \leq c F(n)$$

$$4\left(\frac{n}{3}\right)^2 \leq c n^2$$

$$\frac{4n^2}{9} \leq c n^2$$

$$c = \frac{1}{2} \Rightarrow \frac{4n^2}{9} \leq \frac{n^2}{2} \Rightarrow T(n) = \Theta(n^2)$$

4. Kodo našumo testavimas



Analitiškai apskaičiuotas sudėtingumas n^2 atitinka per našumo testus gautus rezultatus.