**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**Modulio P170B400 „Algoritmų sudarymas ir analizė“**

**Laboratorinio darbo aprašas (ataskaita)**

**Pirmas laboratorinis darbas**

**Dėstytojas**

Lekt. Dalius Makackas

Lekt. Tadas Kraujalis

**Studentas**

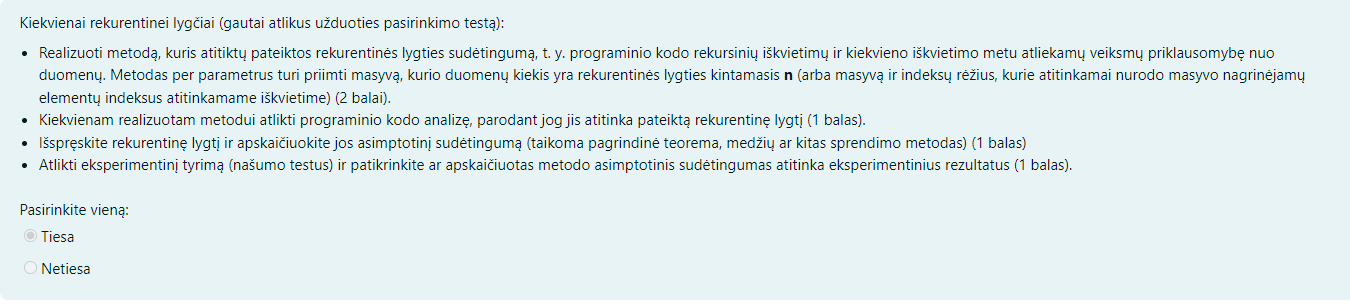
Martynas Kuliešius IFF-1/9

**KAUNAS, 2022**

Turinys:

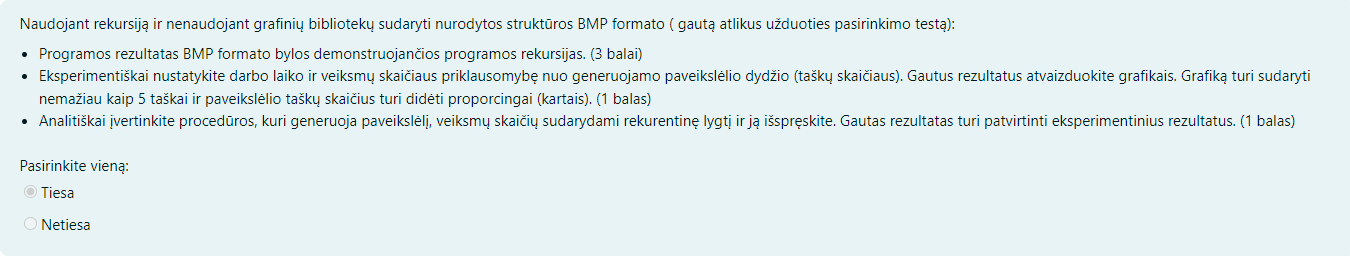
Contents

1. Darbo užduotis
   1. Pirma dalis

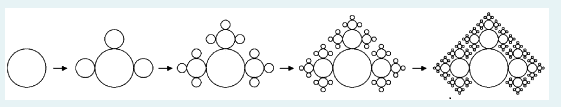


Pirmai užduočiai reikia parašyti metodus trims rekurentinių (rekursinių) lygtims. Gautos lygtys:

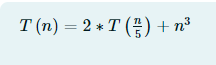
* 
* 
* 
  1. Antra dalis



Šiai užduočiai atlikti reikia parašyti programinį kodą, kuris išvestų .BMP/.PNG formato paveiksliuką pagal duotą fraktalo pavyzdį. Gautas pavyzdys:



1. Pirmoji dalis
   1. Pirma lygtis



Pirmoji lygtis nurodo, kad rekurentinės funkcijos iškvietimai turi būti 2, ir kiekvieną kartą apdirbamas duomenų kiekis sumažinamas 5 kartus. Rekursija kartojasi iki kol n skaičius yra didesnis nei 5. n3 parodo, kad metode turi būti ciklas cikle cikle. Patikrinti veikimą šio metodo, iš duomenų sąrašo gaunamas n-tąsias reikšmes sudėsiu vieną su kita.

Programinis C# kodas aprašyti lygčiai:

static void T1(int n, List<int> TData, ref int number)

{

if (n < 5)

{

return;

}

for(int i = 0;i < n; i++)

{

for (int j = 0; j < n; j++)

{

for (int k = 0; k < n; k++)

{

number += TData[k];

}

}

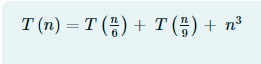
}

T1(n / 5, TData, ref number);

T1(n / 5, TData, ref number);

}

* 1. Antra lygtis



Antroji lygtis nurodo, kad rekurentinės funkcijos iškvietimai turi būti 2, ir kiekvieną kartą apdirbamas duomenų kiekis sumažinamas 6 kartus ir 9 kartus atitinkamai. Rekursija kartojasi iki kol n skaičius yra didesnis nei 6 ir 9. n3 parodo, kad metode turi būti ciklas cikle cikle. Patikrinti veikimą šio metodo, iš duomenų sąrašo gaunamas n-tąsias reikšmes sudėsiu vieną su kita.

Programinis C# kodas aprašyti lygčiai:

static void T2(int n, List<int> TData, ref int number)

{

if (n < 6 || n < 9)

{

return;

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

for (int k = 0; k < n; k++)

{

number += TData[k];

}

}

}

T2(n / 6, TData, ref number);

T2(n / 9, TData, ref number);

}

* 1. Trečia lygtis



Trečioji lygtis nurodo, kad rekurentinės funkcijos iškvietimai turi būti 2, ir kiekvieną kartą apdirbamas duomenų kiekis sumažinamas 8 vienetais ir 2 vienetais atitinkamai. Rekursija kartojasi iki kol n skaičius yra didesnis nei 8 ir 2. n parodo, kad metode turi būti ciklas. Patikrinti veikimą šio metodo, iš duomenų sąrašo gaunamas n-tąsias reikšmes sudėsiu vieną su kita.

Programinis C# kodas aprašyti lygčiai:

static void T3(int n, List<int> TData, ref int number)

{

if (n < 6 || n < 9)

{

return;

}

for (int i = 0; i < n; i++)

{

number += TData[i];

}

T3(n -8, TData, ref number);

T3(n -2, TData, ref number);

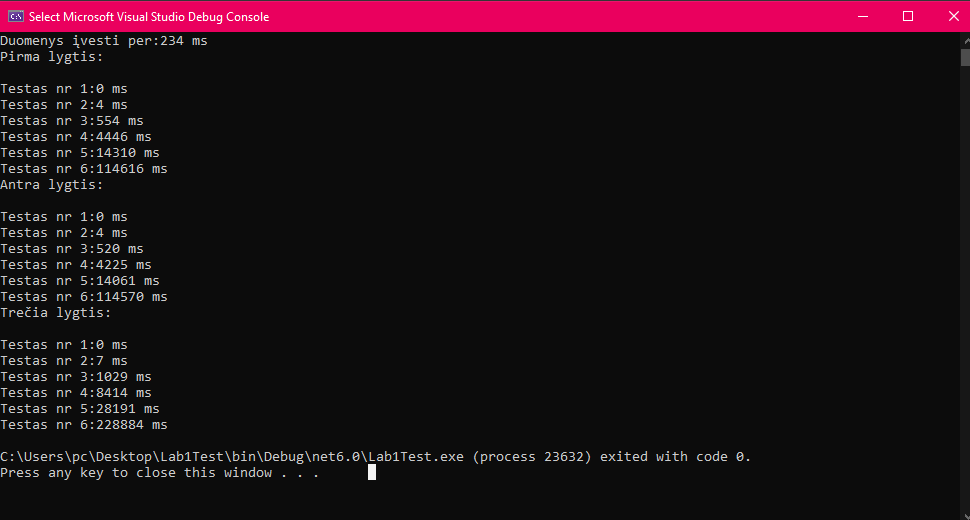
}

1. **Pirmosios dalies našumo testų atlikimas**

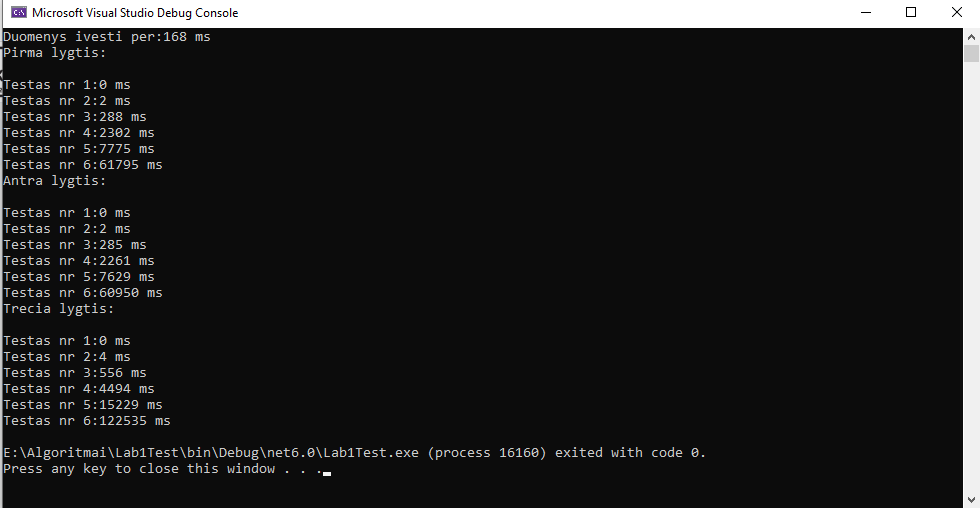
Testuoti rekurentinių lygčių veikimo greičius su mockaroo.com pagalba pasidariau 300000 duomenų .csv failą.

Pervertinęs mano asmeninio, bei universiteto kompiuterių gebėjimus, testuoti su tokiu duomenų kiekiu nepavyko. Testus atlikau naudodamas 50, 100, 500, 1000, 1500, 3000 duomenų.

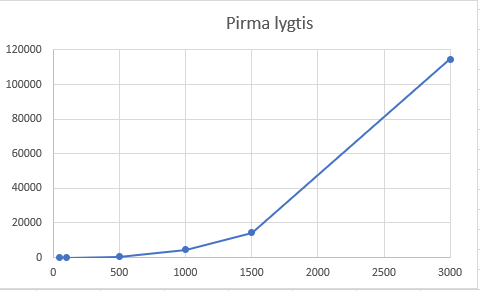
Atlikto testo su asmeniniu kompiuteriu rezultatai:



Atlikto testo su universiteto kompiuteriu rezultatai:

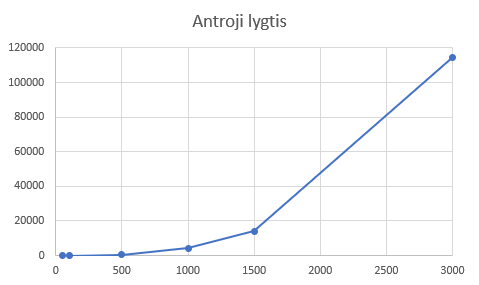


Pirmosios lygties našumo lentelė:



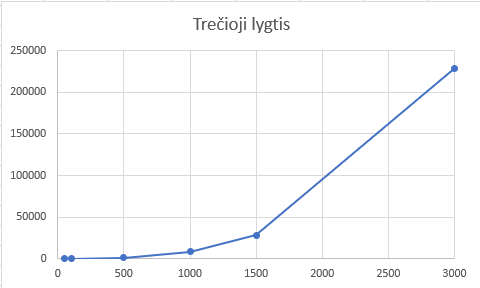
Pirmosios lygties sudėtingumas yra O(n3 \* log n)

Antrosios lygties našumo lentelė:



Antrosios lygties sudėtingumas yra O(n3)

Trečiosios lygties našumo lentelė:



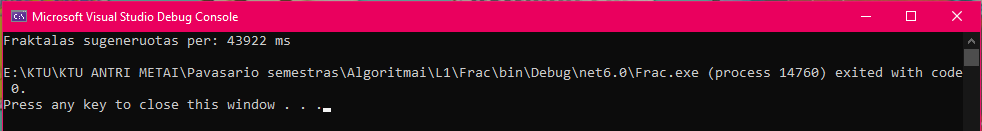
Trečiosios lygties sudėtingumas yra O(n)

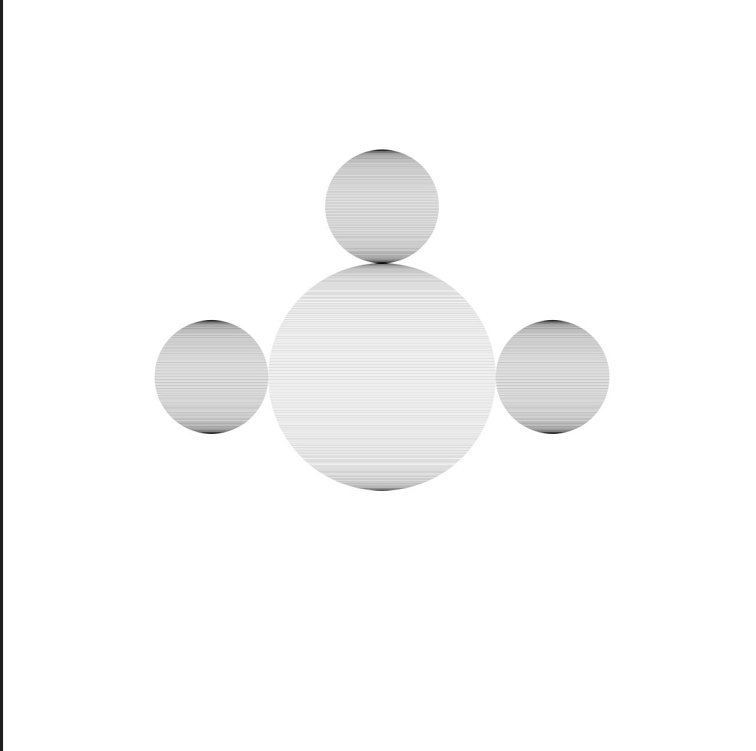
1. Antroji dalis

Antrajai daliai reikėjo parašyti programą, kuri išves .BMP arba .PNG paveiksliuką, pagal duotą fraktalo pavyzdį.

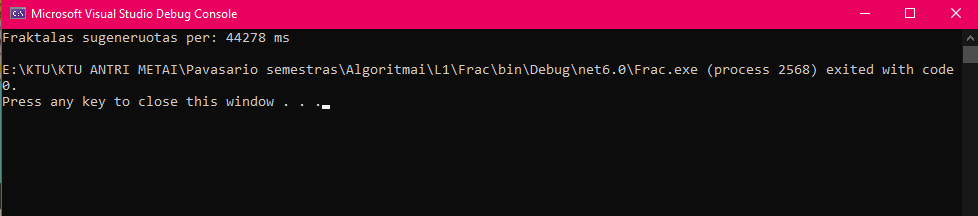
Sugeneravus paveiksliuką, programa parašo, per kiek laiko atliko operaciją. Atlikau 3 testus:

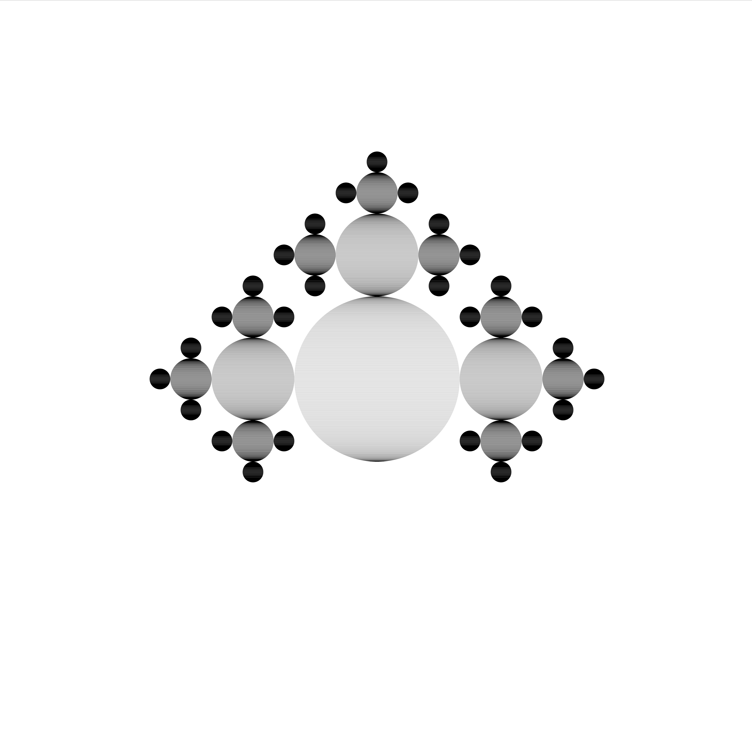
Kai iteruoja 2 kartus:





Kai iteruoja 4 kartus:





Kai iteruoja 6 kartus:

