

**KAUNO TECHNOLOGIJOS**

**UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**ALGORITMŲ SUDARYMAS IR ANALIZĖ**

**P170B400**

**Individualus darbas**

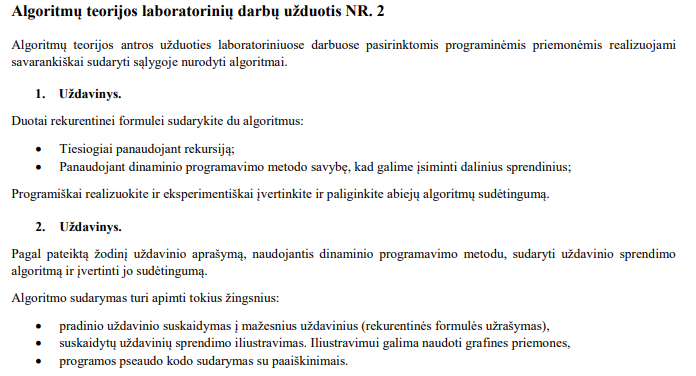
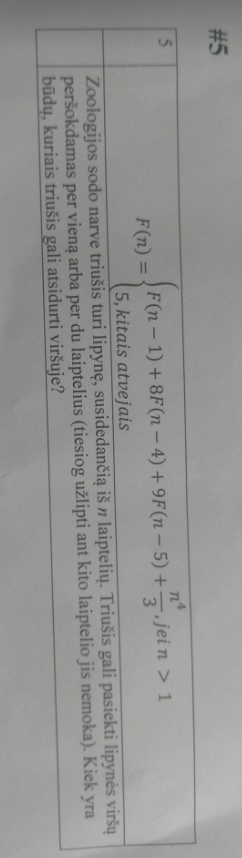
**Atliko:**

Tadas Laurinaitis IFF-6/8

**Priėmė:**

doc. Mikuckienė Irena

Užduotis



Užduotis nr. 1

Rekursija

Kiekis

Kaina

public static double Rekursija(int n)

{

double sum = 0;

if (n > 1)

{

C1

1

sum += Rekursija(n - 1);

1

C2

sum += 8 \* Rekursija(n - 4);

C3

sum += 9 \* Rekursija(n - 5);

1

sum += (Math.Pow(n, 4))/3;

C4

n

return sum;

}

else

{

return 5;

}

}

Sudėtingumas: T = O(n\*n), kadangi n rekursija kartojasi n kartų.

Dinaminis Programavimas

Kaina

Kiekis

public static double DinamineRekursija(double[] p, int n)

{

if(n <= 1)

{

return 5;

}

if(p[n] > 0)

{

return p[n];

}

else if(p[n] < 0)

{

double sum = 0;

sum += DinamineRekursija(p, n - 1);

C1

n - 1

sum += 8 \* DinamineRekursija(p, n - 4);

n - 4

sum += 9 \* DinamineRekursija(p, n - 5);

C2

sum += (Math.Pow(n, 4)) / 3;

n - 5

p[n] = sum;

C3

return sum;

}

return 0;

}

Sudėtingumas: T = O(3n – 10)

|  |  |  |
| --- | --- | --- |
|  | rekursija | dinaminis |
| kiekis | laikas | laikas |
| 5 | 0.0000027 | 0.000001 |
| 10 | 0.0000064 | 0.0000017 |
| 15 | 0.0000703 | 0.000003 |
| 20 | 0.0003788 | 0.0001829 |
| 25 | 0.00154 | 0.00073 |
| 30 | 0.0116 | 0.001572 |
| 35 | 0.0827 | 0.00801 |

Užduotis nr. 2

Užduotis buvo išspręsta naudojantis koreguota Fibonacci algoritmo versija, kurioje žingsniai buvo 2 ir 3.

F(n) = F(n-2) + F(n-3), kadangi F(1) = 0, F(2) = 1, F(3) = 1, F(4) = 1, kadangi esant šiam pakopų skaičiui, galimas tik vienas kelias užlipti. Kitais atvejais triušis žingsniuoja kas 2 ar 3 pakopas.

Kiekis

Kaina

public static double SokantisTriusis(double[] p, int n)

{

double sum = 0;

if (n == 2 || n == 3 || n == 4)

{

return 1;

}

else if(n == 1)

{

return 0;

}

else if(p[n] > 1)

{

return p[n];

}

else

{

(n – 2) + (n-3)

C1

sum += SokantisTriusis(p, n - 2) + SokantisTriusis(p, n - 3);

p[n] = sum;

return sum;

}

}

Sudėtingumas: T = O(N2-5N+6)

|  |  |
| --- | --- |
| 2 užduotis | |
| kiekis | laikas |
| 0 | 0 |
| 5 | 0.0000006 |
| 10 | 0.0000006 |
| 15 | 0.000001 |
| 20 | 0.0000013 |
| 25 | 0.0000017 |
| 30 | 0.000002 |
| 35 | 0.0000023 |

Išvados: Pirmu atvėju rekursinis algoritmas veikė lėčiau su bet kokiu duomenų kiekiu. Tai lėmė didesnis kreipimosi į rekursiją kiekis, bei skirtingas sudėtingumas O(n\*n) ir O(3n - 10);  
Antru atvėju dinaminis algoritmas leido gauti rezultatus daug greičiau, negu tai būtų leidęs sprendimas, naudojantis rekursiją.