**Ministerul Educației și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**



Raport

Lucrarea de laborator nr. 1

Disciplina: **Analiza si proiectarea algortimilor**

Tema: **Algoritmul lui Fibonacci**

|  |  |  |
| --- | --- | --- |
| **A efectuat:** | Student grupa TI-231 FR | Apareci Aurica |
| **A verificat:** | Asistent universitar | Andrievschi-Bagrin Veronica |

**Chișinău 2025**

**Cuprins**

[1.Cadru teoretic 3](#_Toc1)

[2. Listingul programului 4](#_Toc2)

[3. Cazuri de testare 5](#_Toc3)

[4. Concluzii 8](#_Toc4)

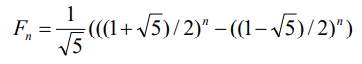
# **Cadru teoretic**

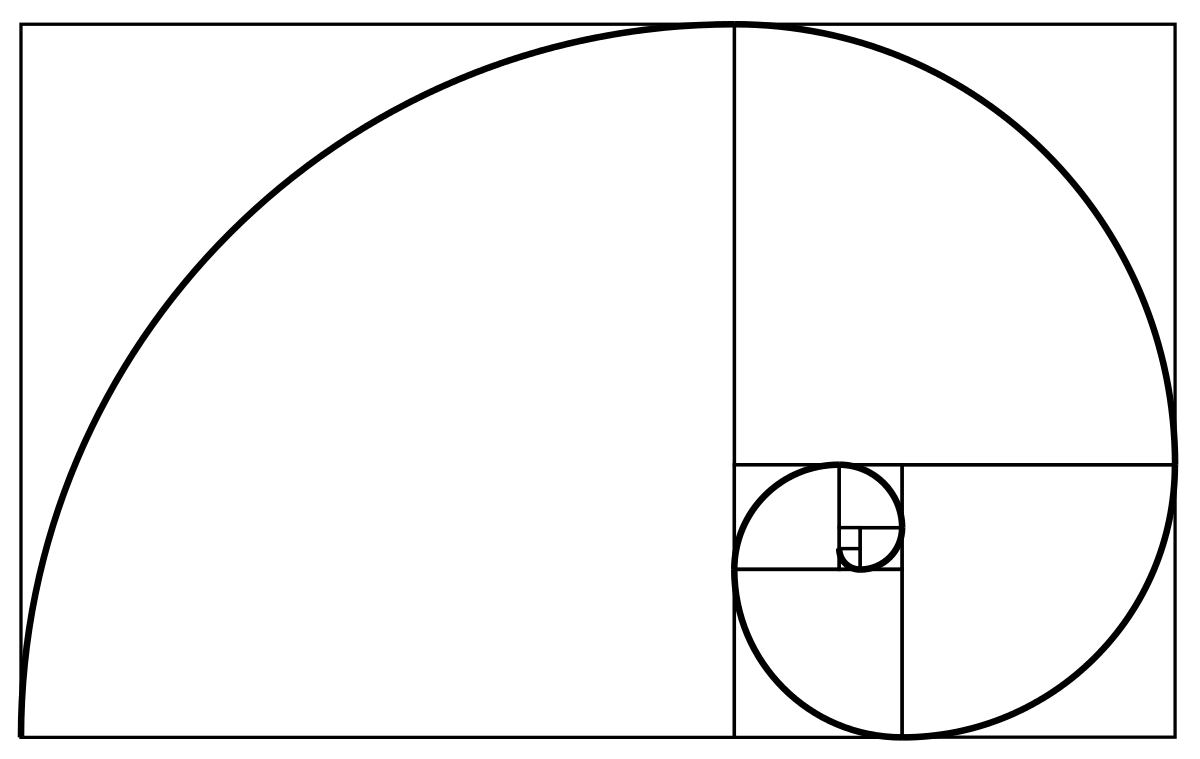
**Tema:**Numerele lui Fibonacci

**Sarcina (conform variantei):**Efectuati analiza empirica a algoritmilor de calculare a numerelor fibonacci. Determinati relatia ce determina complexitatea temporala pentru acesti algortmi.

**Relatia lui Fibonacci** constituie un caz des intalnit in practica. Ea este o relatie de recurenta omogena de ordinul doi cu coeficienti constanti.







# **2. Listingul programului**

**//se utilizeaza urmatoarea biblioteca: GitHub - dotnet/BenchmarkDotNet: Powerful .NET library for benchmarking versiunea: 0.13.2**

**using BenchmarkDotNet.Attributes;**

**using BenchmarkDotNet.Running;**

**namespace lab1v2**

**{**

**internal class Program**

**{**

**static void Main(string[] args)**

**{**

**BenchmarkRunner.Run<Benchy>();**

**}**

**}**

**[MemoryDiagnoser]**

**public class Benchy**

**{**

**int value = 13;**

**[Benchmark]**

**public void fib1()**

**{**

**fibRec(value);**

**}**

**long fibRec(int a) //recursiva**

**{**

**if (a == 0)**

**return 0;**

**if (a == 1)**

**return 1;**

**return fibRec(a - 1) + fibRec(a - 2);**

**}**

**[Benchmark]**

**public void fib2()**

**{**

**fibIt(value);**

**}**

**long fibIt(int a) //iterativa**

**{**

**long f1 = 0;**

**long f2 = 1;**

**long f3 = 1;**

**for (int i = 0; i < a; i++)**

**{**

**f3 = f1 + f2;**

**f1 = f2;**

**f2 = f3;**

**}**

**return f1;**

**}**

**[Benchmark]**

**public void fib3()**

**{**

**fibDir(value);**

**}**

**double fibDir(int a)**

**{**

**return (1 / Math.Sqrt(5)) \* (Math.Pow(((1 +**

**Math.Sqrt(5)) / 2), a) - (Math.Pow(((1 -**

**Math.Sqrt(5)) / 2), a)));**

**}**

**}**

**}**

# **3. Cazuri de testare**

Drept cazuri de testare se vor calcula termenii sirului Fibonnaci cu pozitiile: 5,7,10,10,13,15,20,25,50,60,70,80,90,100. Definim urmatoarele functii ce calcuelaza numerele fibonaci utilizand metode diferite:

Fib1() – algoritm recursiv

Fib2() – algoritm iterativ

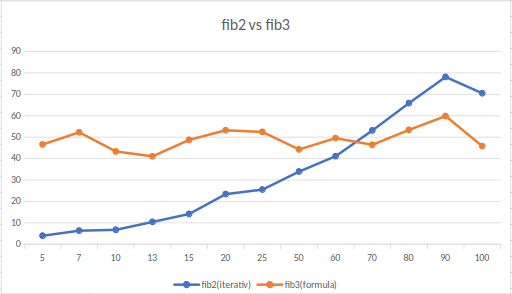
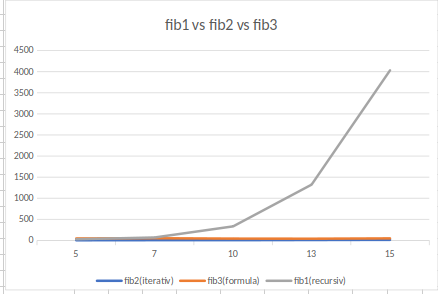
Fib3() – calcul prin formula definita.

|  |
| --- |
| N = 5 |
| N = 7 |
| N = 10 |
| N = 13 |
| N = 15 |
| N = 20 |
| N = 25 |
| N = 50 |
| N = 60 |
| N = 70 |
| N = 80 |
| N = 90 |
| N = 100 |
|  |

**Tabelul valorilor obtinute**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 5 | 7 | 10 | 13 | 15 | 20 | 25 | 50 | 60 | 70 | 80 | 90 | 100 |
| fib2  (iterativ) | 3.95 | 6.33 | 6.72 | 10.41 | 14.12 | 23.39 | 25.5 | 33.92 | 41.13 | 53.11 | 65.89 | 78.12 | 70.5 |
| fib3  (formula) | 46.6 | 52.3 | 43.3 | 41.05 | 48.73 | 53.22 | 52.44 | 44.26 | 49.54 | 46.41 | 53.38 | 59.85 | 45.79 |
| fib1  (recursiv) | 27.8 | 70.1 | 334 | 1322.6 | 4031.8 | 48701 | 488967 |  |  |  |  |  |  |

**Analiza valorilor obtinute**



# **4. Concluzii**

Din cauza neeficientei sale, metoda recursiva a fost exclusa din cercetare dupa calculul celui de al 25-lea numar Fib. In cazul metodei de calcul prin aplicarea algoritmului iterativ, observam o continua crestere a timpului necesar pentru efentuarea operatiilor. Metoda de calcul prin formula poate fi considerata optimala din cauza timpului de executie relativ constant. Potrivit datelor obținute, la calculul celui de al 68-lea număr Fibonacci, metoda iterativa va avea nevoie de ~ aceiași timp ca metoda de calcul prin aplicarea formulei, însă aceste date pot să difere de la mașină de calcul la alta.