НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО»

ЗВІТ

З лабораторної роботи №2

з навчальної дисципліни

«Програмцвання алгоритмічниї структур»

Варіант № 7

Тема:

СКЛАДНІСТЬ АЛГОРИТМУ

Студентки 1-го курсу НН ІАТЕ гр. ТР-22

Грищук Анжеліка

Лабораторну роботу захищено

З оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КИЇВ 2023

**Мета:** Узагальнити на практиці поняття рекурсії та прямої і не прямої рекурсії.

**Код програми:**

#include <iostream>

#include <chrono>

using namespace std;

using namespace chrono;

double sum\_recursive(int n);

double sum\_iterative(int n);

//Обчислення за допомогою рекурсії

double sum\_recursive(int n) {

if (n == 1.0) {

return 0.0;

}

else {

return sum\_recursive(n - 1) + (3.0 \* n) / (n - 1) - n \* n;

}

}

//Обчислення за допомогою циклу

double sum\_iterative(int n) {

double sum = 0.0;

for (int i = 2; i <= n; i++) {

sum += (3.0 \* i) / (i - 1) - i \* i;

}

return sum;

}

int main() {

int sum = 0;

int n;

std::cout << "Enter a value for 'n' --> ";

cin >> n;

auto start\_time = std::chrono::steady\_clock::now();

std::cout << "\nSum recursive = " << sum\_recursive(n) << std::endl;

auto end\_time = std::chrono::steady\_clock::now();

auto time\_recursive = duration\_cast<nanoseconds>(end\_time - start\_time).count();

cout << "Time: " << time\_recursive << " ns" << std::endl;

start\_time = std::chrono::steady\_clock::now();

std::cout << "\nSum non-recursive = " << sum\_iterative(n) << std::endl;

end\_time = std::chrono::steady\_clock::now();

auto time\_iterative = duration\_cast<nanoseconds>(end\_time - start\_time).count();

cout << "Time: " << time\_iterative << " ns" << endl;

return 0;

}

**Формула складності алгоритму обчислення за допомогою циклу в О-нотації О(n)**

**Таблиця часу виконання алгоритму:**

|  |  |  |
| --- | --- | --- |
| **t** | **Кількість кроків** | **Час виконання (нс)** |
| **100** | **100** | **380700** |
| **1000** | **1000** | **784900** |
| **2000** | **2000** | **433800** |
| **3000** | **3000** | **494200** |

**Графік складності алгоритму:**

**Формула складності алгоритму обчислення за допомогою рекурсивної функції в О-нотації О(n)**

**Таблиця часу виконання алгоритму:**

|  |  |  |
| --- | --- | --- |
| **t** | **Кількість кроків** | **Час виконання (нс)** |
| **100** | **100** | **855100** |
| **1000** | **1000** | **860400** |
| **2000** | **2000** | **1458900** |
| **3000** | **3000** | **1549060** |

**Висновок:** Під час виконання лабораторної роботи, я розробила програми згідно з алгоритмом з використанням рекурсивної функції та без використання рекурсивної функції. Оцінила час виконання та складність аглоритму.