Міністерство освіти і науки України

НТУУ «КПІ ім. Ігоря Сікорського»

Лабораторна робота №3

**«РЕКУРСІЯ, ЧАСОВА СКЛАДНІСТЬ**»

Виконала:

Студентка 2-ого курсу ТЕФ

Групи ТВ-91

Таратонова Олександра

Перевірив:

Кузьменко І. М.

КИЇВ 2021

Код програми

**package** pack;

**import** java.util.Random;

**public** **class** main {

**static** **int** *n* = 15000;

**static** **int** *array*[] = **new** **int**[*n*]; //масив випадкових чисел

**public** **static** **void** main(String[] args) {

*generator*(**true**); //генеруємо випадкові числа. Якщо передається true, то числа всі від'ємні

*display*();

**long** time1 = System.*currentTimeMillis*();

System.***out***.println("\nRecursive sum of positive elements: " + *recursive*(0)); //рекурсивний алгоритм

System.***out***.println((**double**)(System.*currentTimeMillis*() - time1));

**long** time2 = System.*currentTimeMillis*();

System.***out***.println("Iterative sum of positive elements: " + *iterative*()); //ітераційний алгоритм

System.***out***.println((**double**)(System.*currentTimeMillis*() - time2));

}

**public** **static** **void** generator(**boolean** flag) {

Random random = **new** Random();

**if**(flag) {

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

*array*[i] = (random.nextInt(1000) \* (-1));

}

}**else** {

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

*array*[i] = (random.nextInt(1000) - 500);

}

}

}

**public** **static** **void** display() {

System.***out***.print("Array: ");

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

System.***out***.print(*array*[i] + " ");

}

}

**public** **static** **int** recursive(**int** i) {

**if**(i == *n* || *array*[i] > 0) {

**return** 0;

}

**return** *array*[i] + *recursive*(i + 1);

}

**public** **static** **int** iterative() {

**int** sum = 0;

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

**if**(*array*[i] > 0) {

**return** sum;

}

sum += *array*[i];

}

**return** sum;

}

}

Для порівняння швидкодії алгоритмів був вимірений час дії при 100, 5000 та 20000 елементів. Після 20000 елементів відбувається переповнення стеку при всіх від’ємних числах. В обох алгоритмах часова складність складає О(n).

Час визначався як середнє значення результатів декількох запусків програми.

|  |  |  |  |
| --- | --- | --- | --- |
| **Всі числа від’ємні** | 100 елементів | 5 000 елементів | 20 000 елементів |
| Час сортування : рекурсивний алгоритм/ ітераційний алгоритм | <1ms/<1ms | 3ms/1ms | 7ms/3ms |
| **Випадковий набір даних** | 100 елементів | 5 000 елементів | 20 000 елементів |
| Час сортування : рекурсивний алгоритм/ ітераційний алгоритм | <1ms/<1ms | <1ms/<1ms | 1ms/0ms |