

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 8 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 15

Виконав студент ІП-15, Костін Вадим Анатолійович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 8

Дослідження алгоритмів пошуку та сортування

Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Варіант 15

Задача

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом.
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом.

№ варіанта	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
15	8 x 5	Дійсний	Із суми додатних значень елементів рядків двовимірного масиву. Відсортувати обміном за зростанням.

Постановка задачі

За умовою задачі дано двовимірний масив. Другий масив буде лічити значення суми додатних елементів кожного рядка. Цей масив відсортуємо методом обміну за зростанням.

Математична модель

Генерувати двовимірний масив А будемо за допомогою підпрограми `inputA()`, яка надає кожному елементу матриці рандомне число від -100 до 100. Другий одновимірний масив В згенеруємо за допомогою підпрограми `inputB()`. Розмірність масиву В дорівнює кількості рядків у масиві А. За допомогою підпрограми `sortedB()` відсортуємо масив В методом обміну.

Змінна	Тип	Ім'я	Призначення
Матриця	Дійсний	А	Початкові дані

Другий масив	Дійсний	B	Результат
Підпрограма що генерує масив A	Процедура	inputA()	Проміжні дані
Підпрограма що генерує масив B	Процедура	inputB()	Результат
Підпрограма що сортує масив B методом обміну за зростанням	Процедура	sortedB()	Результат
Лічильник	Цілочисельний	i	Проміжні дані
Лічильник	Цілочисельний	j	Проміжні дані
Сума додатних елементів рядка	Дійсний	sum	Проміжні дані
Буфер обміну	Дійсний	c	Проміжні дані

Для виразу $x = x + y$ будемо використовувати $x += y$

Для виразу $x = x + 1$ будемо використовувати $x++$

Для надання випадкового значення елементу будемо використовувати $\text{random}(-100, 100)$

Крок 1 Деталізуємо основні дії

Крок 2 Деталізуємо дію генерації масиву A

Крок 3 Деталізуємо дію знаходження суми додатних елементів кожного рядка та створення масиву B

Крок 4 Деталізуємо дію сортування масиву B методом обміну

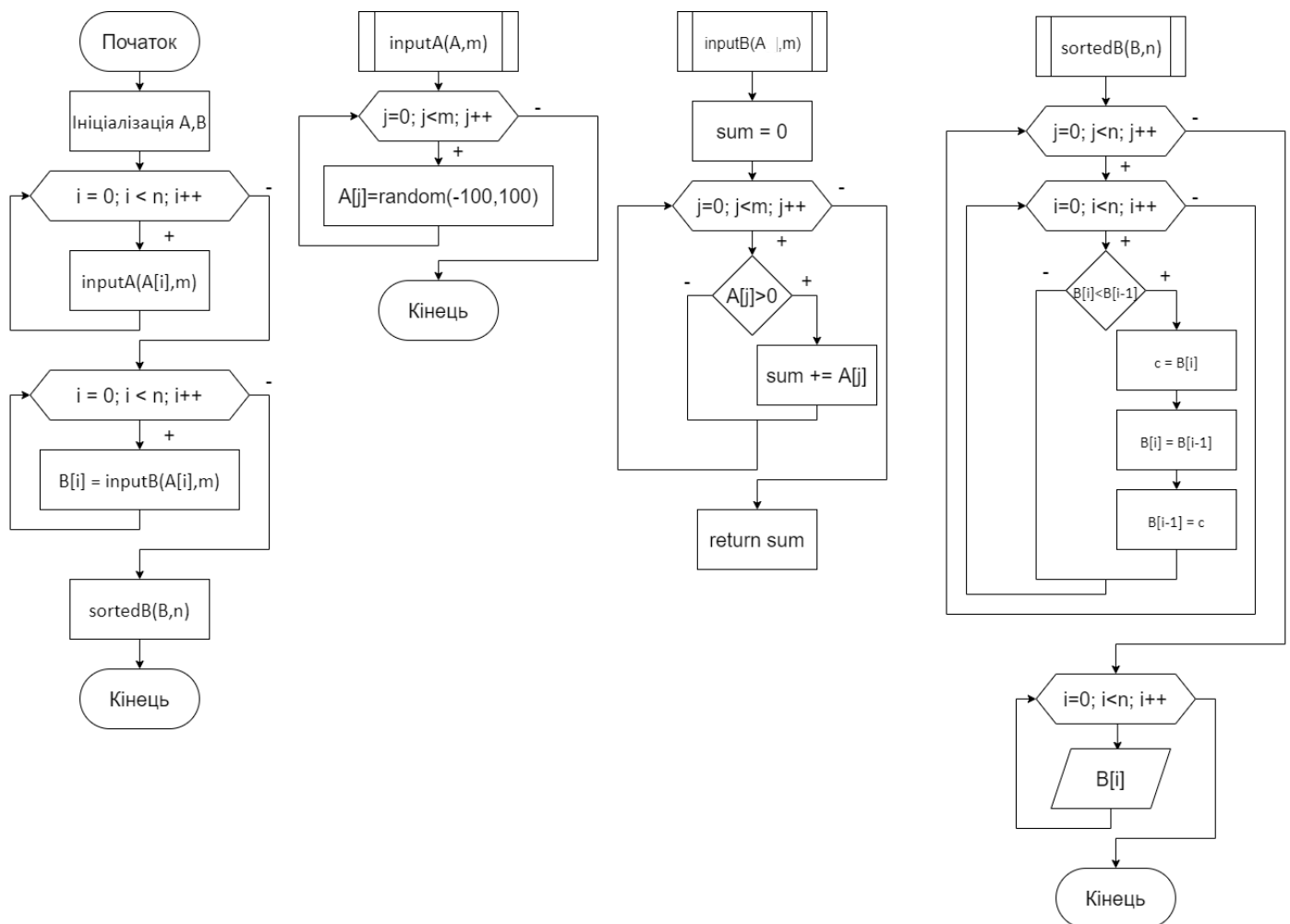
Псевдокод

<p><i>Крок 1</i></p> <p>Початок</p> <p>Ініціалізація A,B</p> <p>Деталізуємо дію генерації масиву A</p> <p>Деталізуємо дію знаходження суми додатних елементів кожного рядка та створення масиву B</p> <p>Деталізуємо дію сортування масиву B методом обміну</p> <p>Кінець</p>	<p><i>Підпрограма inputA()</i></p> <p><i>Підпрограма inputB()</i></p> <p><i>Підпрограма sortedB()</i></p>
---	---

<p><i>Крок 2</i></p> <p>Початок</p> <p>Ініціалізація A, B</p> <p>Повторити для i від 0 до n $inputA(A[i], m)$</p> <p>Все повторити</p> <p>Деталізуємо дію знаходження суми додатних елементів кожного рядка та створення масиву B Деталізуємо дію сортування масиву B методом обміну</p> <p>Кінець</p>	<p><i>Підпрограма $inputA(A, m)$</i></p> <p>Початок</p> <p>Повторити для j від 0 до m $A[j] = random(-100, 100);$</p> <p>Все повторити</p> <p>Кінець</p> <p><i>Підпрограма $inputB()$</i> <i>Підпрограма $sortedB()$</i></p>
<p><i>Крок 3</i></p> <p>Початок</p> <p>Ініціалізація A, B</p> <p>Повторити для i від 0 до n $inputA(A[i], m)$</p> <p>Все повторити</p> <p>Повторити для i від 0 до n $B[i] = inputB(A[i], m)$</p> <p>Все повторити</p> <p>Деталізуємо дію сортування масиву B методом обміну</p> <p>Кінець</p>	<p><i>Підпрограма $inputA(A, m)$</i></p> <p>Початок</p> <p>Повторити для j від 0 до m $A[j] = random(-100, 100);$</p> <p>Все повторити</p> <p>Кінець</p> <p><i>Підпрограма $inputB(A, m)$</i></p> <p>Початок $sum = 0$</p> <p>Повторити для j від 0 до m Якщо $A[j] > 0$ $sum += A[j]$</p> <p>Все якщо</p> <p>Все повторити</p> <p>Повернути sum</p> <p><i>Підпрограма $sortedB()$</i></p>
<p><i>Крок 4</i></p> <p>Початок</p> <p>Ініціалізація A, B</p> <p>Повторити для i від 0 до n $inputA(A[i], m)$</p> <p>Все повторити</p> <p>Повторити для i від 0 до n $B[i] = inputB(A[i], m)$</p> <p>Все повторити $sortedB(B, n)$</p> <p>Кінець</p>	<p><i>Підпрограма $inputA(A, m)$</i></p> <p>Початок</p> <p>Повторити для j від 0 до m $A[j] = random(-100, 100);$</p> <p>Все повторити</p> <p>Кінець</p> <p><i>Підпрограма $inputB(A, m)$</i></p> <p>Початок $sum = 0$</p> <p>Повторити для j від 0 до m Якщо $A[j] > 0$ $sum += A[j]$</p> <p>Все якщо</p> <p>Все повторити</p> <p>Повернути sum</p> <p><i>Підпрограма $sortedB(B, n)$</i></p> <p>Початок</p> <p>Повторити для j від 0 до n Повторити для i від 1 до n Якщо $B[i] < B[i-1]$ $c = B[i]$ $B[i] = B[i-1]$ $B[i-1] = c$</p>

Все повторити
 Все повторити
 Повторити для i від 0 до n
 Вивід $B[i]$
 Все повторити
 Кінець

Блок-схема



Код програми

```

#include <iostream>
#include <ctime>
#include <iomanip>

using namespace std;

void inputA(double *, int);

double inputB(double *, int);
  
```

```

void sortedB(double *, int);

int main(){
    srand(time(NULL));
    int n = 8, m = 5;
    double A[n][m], B[n];

    for (int i = 0; i < n; i++)
    {
        inputA(A[i],m);
    }

    for (int i = 0; i < n; i++)
    {
        B[i] = inputB(A[i],m);
    }

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cout << setw(3) << A[i][j] << " ";
        }
        cout << "| " << B[i] << endl;
    }

    sortedB(B,n);

    system("pause");
}

void inputA(double * a, int N){
    for (int i = 0; i < N; i++)
    {
        a[i] = rand()%200 - 100;
    }
}

double inputB(double * a, int n){
    double sum = 0;
    for (int i = 0; i < n; i++)
    {
        if(a[i]>0)
            sum += a[i];
    }
    return sum;
}

void sortedB(double * b, int n){
    for (int j = 0; j < n; j++)
    {
        for (int i = 1; i < n; i++)
        {

```

```

        if(b[i]<b[i-1]){
            int c = b[i];
            b[i] = b[i-1];
            b[i-1] = c;
        }
    }
}
cout << endl << endl;
for (int i = 0; i < n; i++)
{
    cout << b[i] << " ";
}
}

```

Результат роботи програми

A

B

32	-40	43	-96	-47		75
75	10	-89	32	16		133
96	45	-56	-86	69		210
6	-36	80	-37	-90		86
98	11	24	-86	68		201
80	-23	-77	57	56		193
-72	-88	4	-34	17		21
-53	27	-47	-38	-24		27

21	27	75	86	133	193	201	210
----	----	----	----	-----	-----	-----	-----

sorted B

Висновки

Протягом восьмої лабораторної роботи ми дослідили алгоритми пошуку та сортування, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій.