

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 лінійних алгоритмів»

Варіант 29

Виконав студент ІП-13 Романюк Діана Олексіївна
(шифр, прізвище, ім'я, по батькові)

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

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 29

Постановка задачі:

Задати матрицю розмірністю 5×8 , знайти мінімальний елемент кожного стовпця, винести у масив та виконати сортування методом Шела за зростанням.

Побудова математичної моделі:

Таблиця імен змінних

Змінна	Тип	Ім'я	Призначення
Двовимірний масив(матриця)	цілий	Array	Проміжні дані
Масив	цілий	Array2	Проміжні дані/вихідні дані
Лічильник	цілий	i	Проміжні дані
Лічильник	цілий	j	Проміжні дані
Мінімальне значення у стовпці	цілий	min	Проміжні дані
Крок у сортуванні	цілий	step	Проміжні дані
Зміна елементів місцями	цілий	change	Проміжні дані

Таблиця імен констант:

Змінна	Тип	Ім'я	Призначення
Кількість рядків	цілий	row	Початкові дані
Кількість стовпчиків	цілий	column	Початкові дані

Таблиця імен підпрограм:

Змінна	Тип	Ім'я	Призначення
Виведення масиву	цілий	Out	Проміжні дані
Виконання сортування Шела	цілий	ShelSort	Вихідні дані

Ініціалізація кількості рядків **row** та стовпчиків **column** двовимірного масиву.

Ініціалізація двовимірного і одновимірного масивів **Array** та **Array2**

srand(time(0)) – функція для анулювання збережених рандомних чисел, яким заповниться двовимірний масив.

rand() – функція для генерації рандомних чисел для заповнення двовимірного масиву.

Array[][] = rand() % 10 – заповнення двовимірного масиву рандомними числами від 1 до 10 за допомогою вкладених циклів. Зовнішній цикл обмежує кількість згенерованих чисел за кількістю рядків, а внутрішній за кількістю стовбчиків. Виведемо отримано матрицю за допомогою циклу для перевірки результату.

Далі будуємо два вкладених цикли для знаходження мінімуму **min**

для $j = 0; j < \text{column}; j++$ - ініціалізуємо змінну $\text{min} = \text{Array}[0][j]$

всередині цикл для $i = 0; i < \text{row}; i++$ - якщо $\text{Array}[i][j] < \text{min}$, то

$\text{min} = \text{Array}[i][j]$

Виходимо з циклу і присвоюємо j-тому елементу масива мінімальне значення відповідного стовпчика

$\text{Array2}[j] = \text{min}$

У функції сортування **ShelSort()** ініціалізуємо змінну **step** – «крок» - тобто число елементів через які буде відбуватися вставка

Будуємо цикл поки крок більше 0, вкладаємо цикл для $i = 0; i < (\text{кількість стовпчиків column} - \text{крок step}); i++$ ініціалізуємо та присвоюємо $j = i$, далі вкладаємо ще один цикл поки $j \geq 0$ та $\text{Array2}[j] > \text{Array2}[j + \text{step}(\text{крок})]$

Тобто йдемо від i-того елементу поки не прийдемо до початку масива і поки розглядаємо елемент, що знаходиться на більшій відстані ніж елемент, що знаходиться на відстані кроку ініціалізуємо змінну **change** та змінюємо елементи місцями

$\text{change} = \text{Array2}[j];$

$\text{Array2}[j] = \text{Array2}[j + \text{step}];$

$\text{Array2}[j + \text{step}] = \text{temp};$

$j--$

виходимо з двох циклів та зменшуємо крок вдвічі

$\text{step} = \text{step} / 2$

Задача:

Задати матрицю розмірністю 5×8 , знайти мінімальний елемент кожного стовпця, винести у масив та виконати сортування методом Шела за зростанням.

Псевдокод алгоритму:

початок

```
row = 5
column = 8
Array[row][column]
Array2[column]
srand(time(0))
повторити для i від 0 до row
    повторити для j від 0 до column
        Array[i][j] = rand() % 10
    все повторити
все повторити
повторити для i від 0 до row
    повторити для j від 0 до column
        вивести Array[i][j]
    все повторити
все повторити
повторити для j від 0 до column
    min = Array[0][j]
    повторити для i від 0 до row
        якщо Array[i][j] < min
            то min = Array[i][j]
    все якщо
    все повторити
    Array2[j] = min
все повторити
Out(column, Array2)
ShelSort(column, Array2)
Out(column, Array2)
```

кінець

Псевдокод підпрограм:

підпрограма

```
Out(column, Array2):
    повторити для i від 0 до column
        вивід Array2[i]
    все повторити
все підпрограма
```

підпрограма

ShelSort(column, Array2):

step = column / 2

повторити

повторити для i від 0 до (column – step)

j = i

повторити

change = Array2[j]

Array2[j] = Array2[j + step]

Array2[j + step] = change

j--

поки j >= 0 && Array2[j] > Array2[j + step]

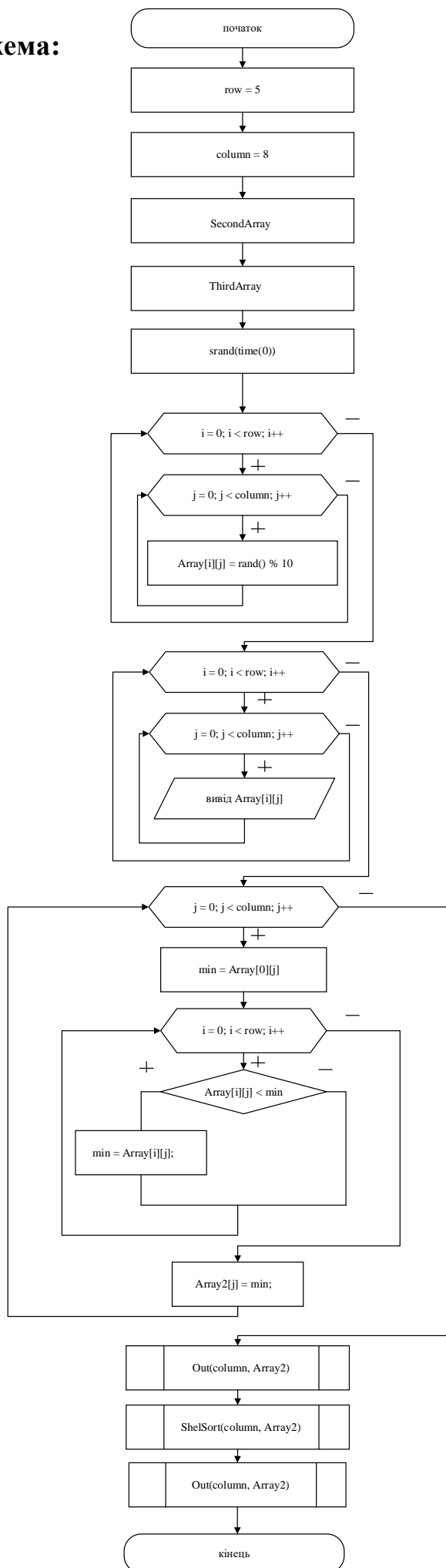
все повторити

step = step / 2

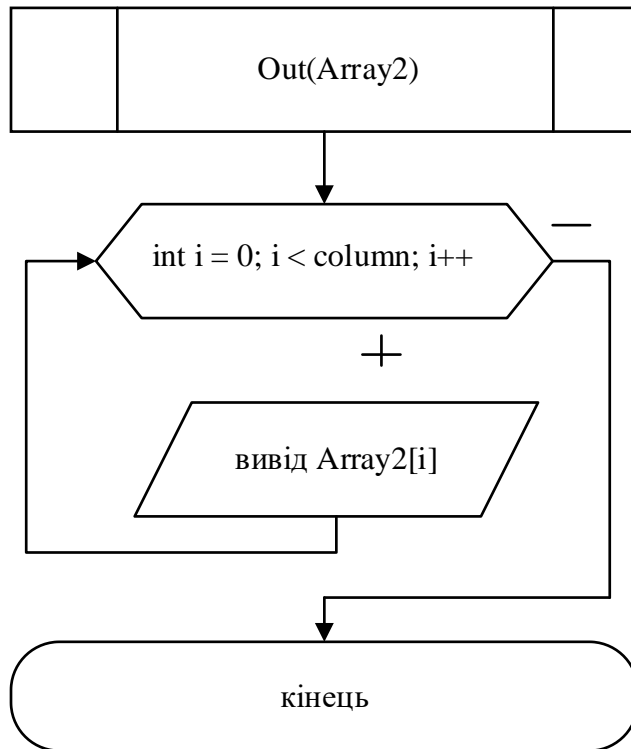
поки j >= 0 && Array2[j] > Array2[j + step]

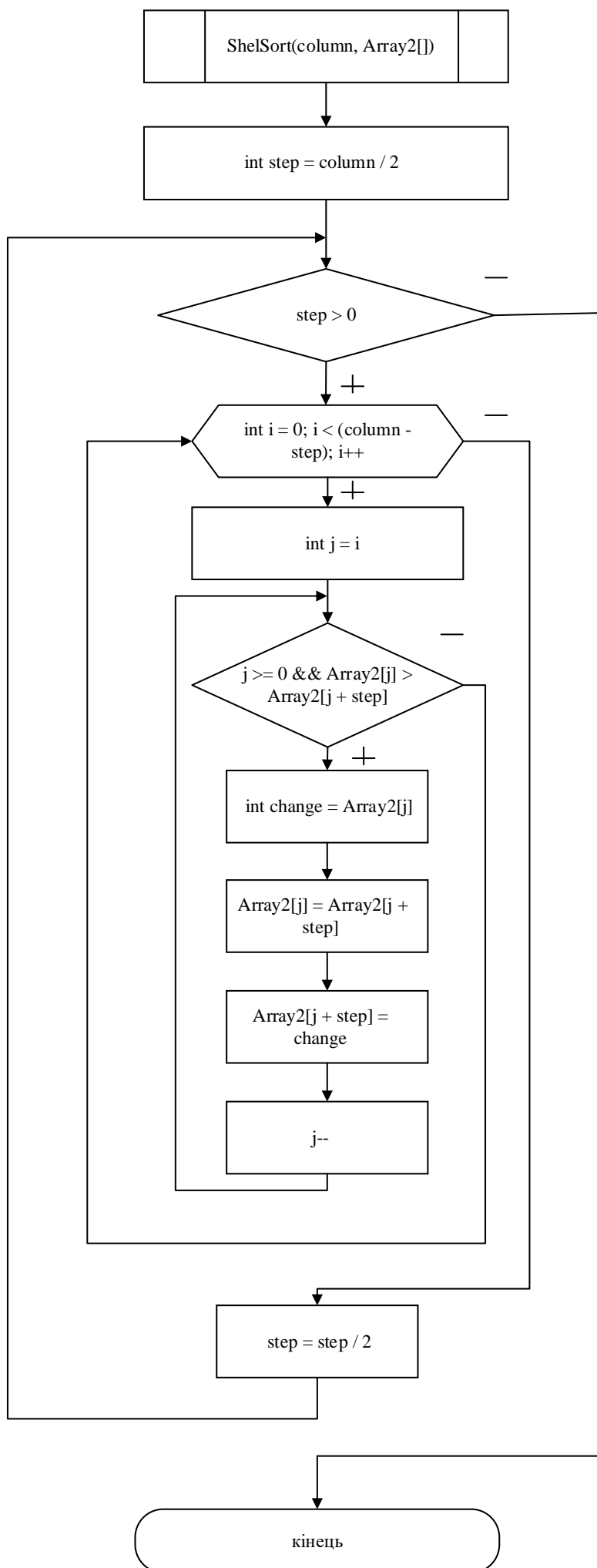
все підпрограма

Блок-схема:



Блок-схема підпрограм:





Код програми:

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

void ShellSort(int column, int Array2[]){
    int step = column / 2;
    while (step > 0){
        for (int i = 0; i < (column - step); i++){
            int j = i;
            while (j >= 0 && Array2[j] > Array2[j + step]){
                int change = Array2[j];
                Array2[j] = Array2[j + step];
                Array2[j + step] = change;
                j--;
            }
            step = step / 2;
        }
    }
}

void Out(int column, int Array2[]){
    cout << "n";
    for (int i = 0; i < column; i++) {
        cout << "(Array2 + i) << " ";
    }
    cout << endl;
}

int main() {
    const int row = 5;
    const int column = 8;
    int Array[row][column];
    int Array2[column];
    srand(time(0));
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            Array[i][j] = rand() % 10;
        }
    }
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            cout << Array[i][j] << " ";
        }
        cout << endl;
    }
    for (int j = 0; j < column; j++) {
        int min = Array[0][j];
        for (int i = 0; i < row; i++) {
            if (Array[i][j] < min) {
                min = Array[i][j];
            }
        }
        Array2[j] = min;
    }
    Out(column, Array2);
    ShellSort(column, Array2);
    Out(column, Array2);
}
```

```
Консоль отладки Microsoft Visual Studio
3 8 6 0 1 3 5 5
6 1 7 1 8 0 0 7
9 4 8 2 3 8 0 3
3 9 2 1 0 3 4 9
7 7 7 9 8 8 7 4

3 1 2 0 0 0 0 3
0 0 0 1 1 2 3 3

Консоль отладки Microsoft Visual Studio
9 9 5 3 6 8 8 6
0 0 4 7 1 5 8 4
1 8 1 2 9 4 2 9
1 3 6 1 3 6 2 9
1 1 5 0 7 4 1 1

0 0 1 0 1 4 1 1
0 0 0 1 1 1 1 4
```

Висновки:

Під час виконання лабораторної роботи досліджено алгоритми пошуку та сортування. Отримано практичні навички їх використання під час складання лінійних програмних специфікацій. Побудовано математичну модель задачі та таблицю імен змінних. Розроблено псевдокод вирішення даної математичної задачі. Умовно розбито виконання коду на кроки, а також описано його виконання за допомогою створення відповідної блок-схеми. Перевірено умовне виконання коду за допомогою випробування алгоритму.