Додаток 1

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

«Дослідження алгоритмів пошуку

та сортування»

Варіант 4

Виконав студент Бутов Даниіл Романович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вечерковська 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 4

Завдання. 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.

№	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
4	4x4	Цілий	Із мінімальних значень елементів стовпців двовимірного масиву. Відсортувати методом вставки за спаданням.

Постановка задачі.

Нам потрібно згенерувати двовимірний масив 4x4 цілого типу, де потрібно знайти мінімальні елементи кожного стовпця. Результатом буде відсортований масив мінімальних значень методом вставки за спаданням.

Побудова математичної моделі.

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Цілий	array	Початкове
Відсортований масив	Цілий	sorted_array	Результат
Розмір масива	Цілий	size	Початкове
Мінімальне значення	Цілий	min	Проміжне
Генерування масиву	Процедура	generate_array	Початкове
Найменьші елементи масиву	Процедура	find_mins()	Проміжні дані
Сортування масиву	Процедура	sort_arr()	Проміжні дані
Вивід масивів	Процедура	our_arr()	Результат

array[4][4] згенеруємо випадковими цілими числами. Потім ми знайдемо мінімальні значення кожного стовбця завдяки вкладеного циклу, де ми будемо перебирати кожну цифру стовпця. Потім ми будемо сортувати за спаданням створений та заданий sorted_array[4]. Завдяки вкладеного циклу ми будемо перебирати числа з масиву та переставляти іх за спадання, для цього ми ініціалізуємо temp, яка буде хранити у собі відсортований перший єлемент (за аналогією, що масив з одним елементов вже є відсортованим) та буде його порівнювати за спаданням з іншими єлементами. Після сортування, ми виведемо масиви завдяки 2 циклів щоб перевірити алгоритм. Також створимо змінну size, яка містить у собі розмір масиву. У функції сортування, ініціалізація i, j було зроблено до цикла. У функції find_mins() ми також створили змінну min, для зберігання мінмального значення.

Розв'язання.

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначемо основні дії.

Крок 2. Деталізуємо генерування масиву array[4][4].

Крок 3. Деталізуємо пошук мінімальний значень та генерування sorted_array[4].

Крок 4. Деталізуємо сортування методом вставки за спаданням.

Крок 5. Деталізуємо вивід масивів.

Псевдокод.

Крок 1.

Початок

Генерування масиву array

Пошук мінімальних значень та генерування sorted_array

Сортування масиву sorted_array

Вивід масивів

Кінець

Крок 2.

Початок

generate_array(array, size)

Пошук мінімальних значень та генерування sorted_array

Сортування масиву sorted_array

Вивід масивів

Кінець

Крок 3.

Початок

generate_array(array, size)

find_mins(array, sorted_array, size)

Сортування масиву sorted_array

Вивід масивів

Кінець

Крок 4.

Початок

generate_array(array, size)

find_mins(array, sorted_array, size)

sort_arr(sorted_array, size)

Вивід масивів

Кінець

Крок 5.

Початок

generate_array(array, size)

find_mins(array, sorted_array, size)

sort_arr(sorted_array, size)a

out_arr(array, sorted_array, size)

Кінець

```
generate_array(array,  size)
Початок
    для i до size повторити
        для j до size повторити
            array[i][j] = rand()% 10;
        все повторити
    все повторити
Кінець
```

```
find_mins(array, sorted_array, size)
Початок
    для i до size повторити
        min = array[0][i]
        sorted_array[i] = min
        для j до size повторити
            якщо min > arr[j][i]
              то
                min = arr[j][i]
                sort[i] = min
            все якщо
        все повторити
    все повторити
Кінець
```

sort_arr(sorted_array, size)

Початок

для i **до** size **повторити**

temp = sort[i]

для j **до** j >= 0 && sort[j] < temp**;** j-- **повторити**

sort[j+1] = sort[j]

все повторити

sort[j+1] = temp

все повторити

Кінець

out_arr(array, sorted_array, size)

Початок

для i **до** size **повторити**

для j **до** size **повторити**

виведення array[i][j]

все повторити

все повторити

для i **до** size **повторити**

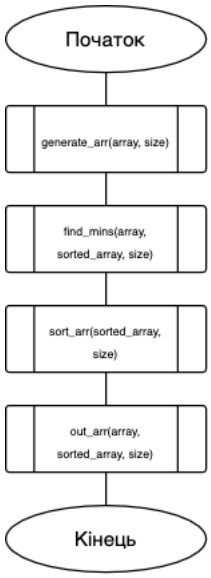
виведення sorted_array[i]

все повторити

Кінець

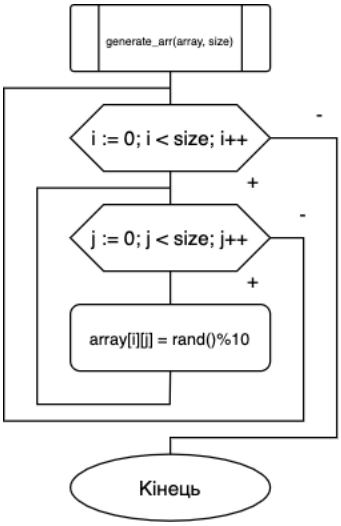
Блоксхема.

Основна програма.

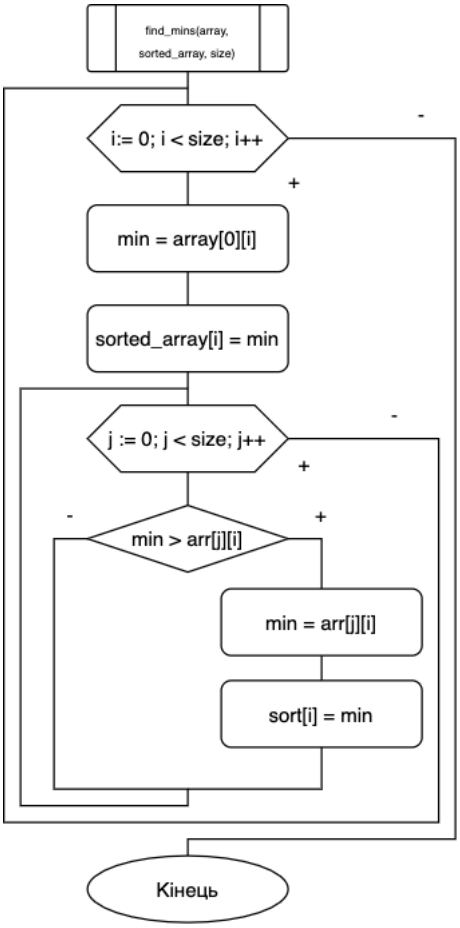


Підпрограми.

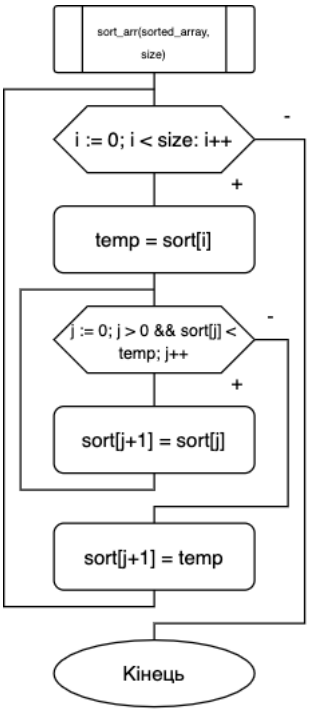
generate_arr



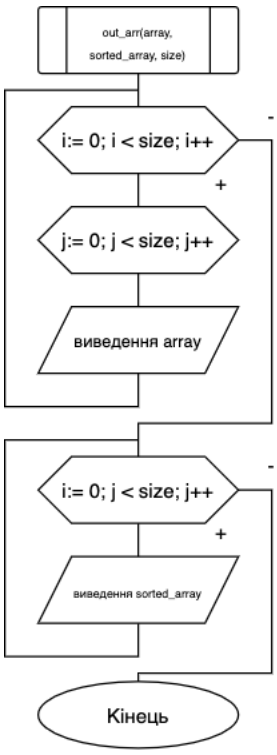
find_mins



sort_arr



out_arr



Код

```cpp
#include <iostream>
using namespace std;

void generate_array(int arr[4][4], int size);
void find_mins(int arr[4][4], int sort[4], int size);
void sort_arr(int sort[4], int size);
void out_arr(int arr[4][4], int sort[4], int size);
int main(){
    int array[4][4];
    int sorted_array[4];
    int size = 4;
    generate_array( arr: array, size);
    find_mins( arr: array,  sort: sorted_array, size);
    sort_arr( sort: sorted_array, size);
    out_arr( arr: array,  sort: sorted_array, size);
}

void generate_array(int arr[4][4], int size){
    srand(time(nullptr));
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            arr[i][j] = rand()% 10;
        }
    }
}

void find_mins(int arr[4][4], int sort[4], int size){
    int min;
    for (int i = 0; i < size; ++i) {
        min = arr[0][i];
        sort[i] = min;
        for (int j = 0; j < size; ++j) {
            if (min > arr[j][i]) {
                min = arr[j][i];
                sort[i] = min;
            }
        }
    }
}

void sort_arr(int sort[4], int size){
    int i , j , temp;
    for (i = 0; i < size; ++i) {
        temp = sort[i];
        for (j = i-1; j >= 0 && sort[j] < temp; j--)
            sort[j+1] = sort[j];
        sort[j+1] = temp;
    }
}

void out_arr(int arr[4][4], int sort[4], int size){
    cout << "Random array is:\n";
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
           cout << arr[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl << "Sorted array with minimal values:\n";
    for (int i = 0; i < size; ++i) {
        cout << sort[i] << " ";
    }
}
```

Тестування.

```
Random array is:
1 7 2 0
7 9 6 2
2 7 7 4
6 5 9 9

Sorted array with minimal values:
5 2 1 0
```

Випробування.

Блок	Дія
	Початок
1	generate_array(array, size)
	1, 7, 2, 0, 7, 9,6 , 2,
	2, 7, 7, 4, 6, 5, 9, 9
2	find_mins(array, sort_array, size)
	1 7 **2 0**
	7 9 6 2
	2 7 7 4
	6 **5** 9 9
	sorted_array = 1,5,2,0
3	sort_arr(sorted_array, size)
	i, j ,temp
	1,5,2,0 --> **5**,1,2,0 --> 5,**2**,1,0 -->
	5,2,1,**0**
	sorted_array = 5,2,1,0
4	Виведення масивів

Висновок.

Ми дослідили алгоритми пошуку та сортування, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій. Розробили алгоритм та склали програму, яка має у собі раціональний алгоритм пошуку та сортування вставками зі спаданням. Також навчилися скаладти алгоритм у якому задіяні двовимірні масиви та на їх основі утворюємо нові масиви.