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 24

Виконав студент ІП-11 Печковський Олександр Костянтинович

(шифр, прізвище, ім'я, по батькові)

Мартинова Оксана Петрівна

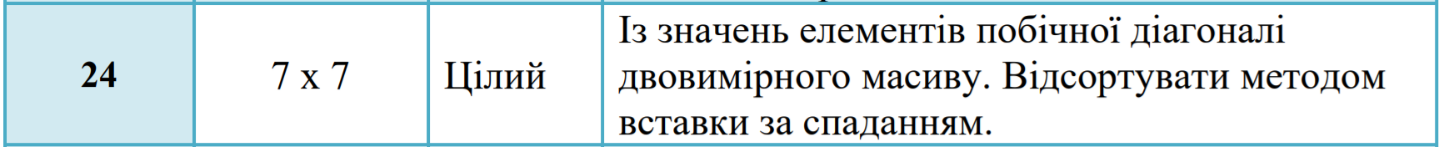
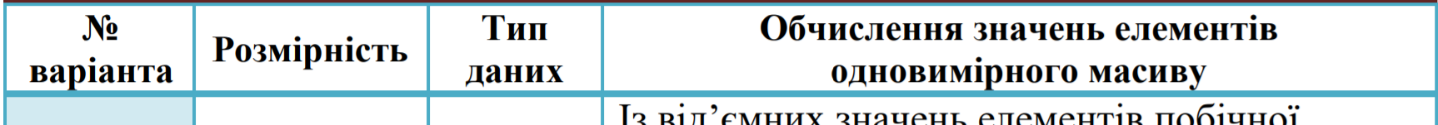
Перевірила

(прізвище, ім'я, по батькові)

Київ 2021

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

**Умова задачі:**



Постановка задачі:

1. Створимо змінну індексованого типу (двовимірний масив) arr[7][7]. Ініціюємо її значеннями, згенерованими випадковим чином.

2. Створимо змінну індексованого типу (одновимірний масив) arr2[7]. Ініціюємо її значеннями елементів побічної діагоналі двовимірного масиву arr[7][7].

3. Відсортуємо елементи масиву arr2[7] методом вставки за спаданням.

Математична модель:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| arr | int[][] | Двовимірний масив | Початкове значення |
| arr2 | int[] | Одновимірний масив | Результат |
| a | const int | Константа, яка дорівнює 7 | Проміжне значення |
| i | int | Лічильник циклу | Проміжне значення |
| j | int | Лічильник циклу | Проміжне значення |
| k | int | Лічильник циклу | Проміжне значення |
| temp | int | Тимчасова змінна | Проміжне значення |

**Псевдокод:**

Крок 1: Визначимо основні дії

Крок 2: Деталізуємо заповнення масиву arr випадковими значеннями і виведення його на екран

Крок 3: Деталізуємо заповнення масиву arr2 значеннями елементів побічної діагоналі масиву arr і виведення його на екран

Крок 4: Деталізуємо сортування елементів масиву arr2 методом вставки за спаданням і виведення його на екран

**Крок 1:**

**Початок**

Заповнення масиву arr випадковими значеннями і виведення його на екран

Заповнення масиву arr2 значеннями елементів побічної діагоналі масиву arr і виведення його на екран

Сортування елементів масиву arr2 методом вставки за спаданням і виведення його на екран

**Кінець**

**Крок 2:**

**Початок**

Для i в діапазоні (0, 7):

Для j в діапазоні (0, 7):

arr[i][j] = (int)(-9 + rand() % 19)

Виводимо arr[i][j]

Заповнення масиву arr2 значеннями елементів побічної діагоналі масиву arr і виведення його на екран

Сортування елементів масиву arr2 методом вставки за спаданням і виведення його на екран

**Кінець**

**Крок 3:**

**Початок**

Для i в діапазоні (0, 7):

Для j в діапазоні (0, 7):

arr[i][j] = (int)(-9 + rand() % 19)

Виводимо arr[i][j]

Для j в діапазоні (0, 7):

Для i в діапазоні (0, 7):

Якщо i + j = a – 1:

arr2[k] = arr[i][j]

k++

Виводимо arr2[j];

Сортування елементів масиву arr2 методом вставки за спаданням і виведення його на екран

**Кінець**

**Крок 4:**

**Початок**

Для i в діапазоні (0, 7):

Для j в діапазоні (0, 7):

arr[i][j] = (int)(-9 + rand() % 19)

Виводимо arr[i][j]

Для j в діапазоні (0, 7):

Для i в діапазоні (0, 7):

Якщо i + j = a – 1:

arr2[k] = arr[i][j]

k++

Виводимо arr2[j]

Для i в діапазоні (1, 7):

j=i

Поки (j > 0 та arr2[j - 1] < arr2[j]):

temp = arr2[j - 1];

arr2[j - 1] = arr2[j];

arr2[j] = temp;

j--

Для i в діапазоні (0, 7):

Виводимо arr2[i]

**Кінець**

**Блок-схеми:**









Код програми:

#include <iostream>

#include "windows.h"

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

const int a = 7;

int arr[a][a];

int arr2[a];

int k = 0;

cout << "Двовимірний масив з рандомними елементами:" << endl;

for (int i = 0; i < a; ++i)

{

for (int j = 0; j < a; ++j)

{

arr[i][j] = (int)(-9 + rand() % 19);

cout.width(5);

cout << arr[i][j];

}

cout << endl;

}

for (int j = 0; j < a; j++)

{

for (int i = 0; i < a; i++)

{

if (i + j == a - 1)

{

arr2[k] = arr[i][j];

k++;

}

}

}

cout << "\nОдновимірний масив з елементів побічної діагоналі:" << endl;

for (int i = 0; i < a; i++)

{

cout.width(5);

cout << arr2[i];

}

for (int i = 1; i < a; i++)

{

int j = i;

while (j > 0 && arr2[j - 1] < arr2[j])

{

int temp = arr2[j - 1];

arr2[j - 1] = arr2[j];

arr2[j] = temp;

j--;

}

}

cout << "\n\nОдновимірний масив, відсортований методом вставки за спаданням:" << endl;

for (int i = 0; i < a; i++)

{

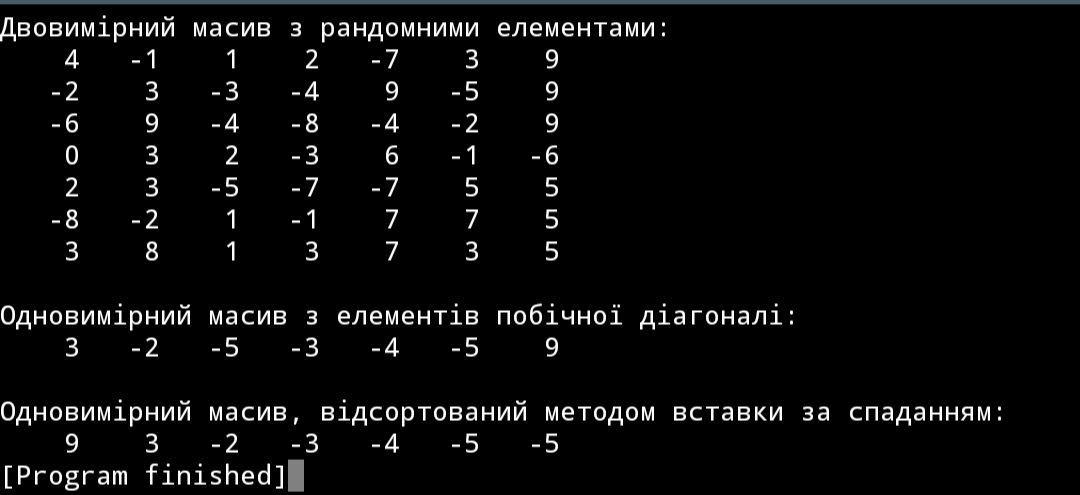
cout.width(5);

cout << arr2[i];

}

}

Робота програми:



Випробування алгоритму:

|  |  |  |
| --- | --- | --- |
|  | Одновимірний масив | Відсортований одновимірний масив |
| 0 | 3 | 9 |
| 1 | -2 | 3 |
| 2 | -5 | -2 |
| 3 | -3 | -3 |
| 4 | -4 | -4 |
| 5 | -5 | -5 |
| 6 | 9 | -5 |

Порівнюючи результати аналітичного сортування і сортування програмою, бачимо, що програма сортує масив правильно.

Висновок: виконуючи лабораторну роботу, я дослідив алгоритми пошуку та сортування, набув практичних навичок використання цих алгоритмів під час складання програмних специфікацій.