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

Лабораторна робота №6

з дисципліни
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-44
Мундурс Нікіта Юрійович
номер у списку групи: 16

Перевірила:

Молчанова А. А.

Київ 2024

Завдання

1. Задано двовимірний масив (матрицю) цілих чисел A[m,n] або A[n,n], де m та n – натуральні числа (константи), що визначають розміри двовимірного масиву. Виконати сортування цього масиву або заданої за варіантом його частини у заданому порядку заданим алгоритмом (методом).

Сортування повинно бути виконано безпосередньо у двовимірному масиві «на тому ж місці», тобто без перезаписування масиву та/або його будь-якої частини до інших одно- або двовимірних масивів, а також без використання спискових структур даних.

2. Розміри матриці m та n взяти самостійно у межах від 7 до 10.

3. При тестуванні програми необхідно підбирати такі вхідні набори початкових значеннь матриці, щоб можна було легко відстежити коректність виконання сортування і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

Варіант 16:

Заддано двовимірний масив (матрицю) цілих чисел $A[m, n]$.
Віддсортувати елементи останнього стовпчика масиву, що стоять на непарних позиціях, алгоритмом №2 методу обмінів («бульбашкове сортування» з використанням «прапорця») за незбільшенням.

Текст програми:

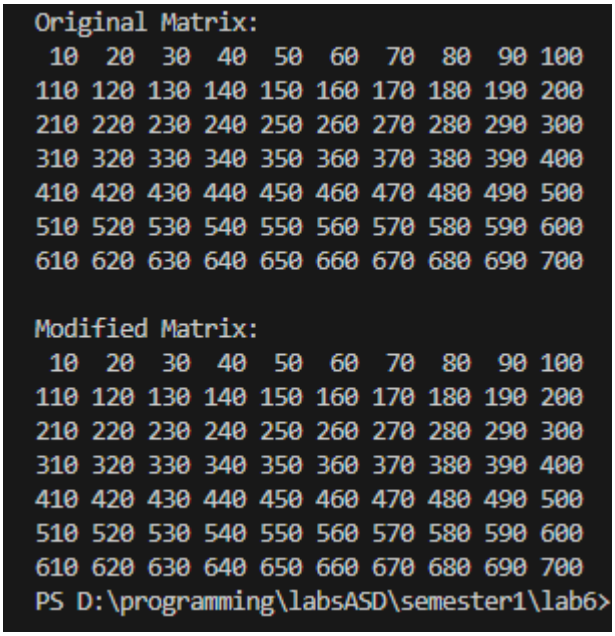
```
#include <stdio.h>


#define ROWS 7

#define COLS 10


void bubbleSortWithFlag(int arr[], int n) {
  int temp;
  int swapped;


  for (int i = 0; i < n - 1; i++) {
    swapped = 0;
    for (int j = 0; j < n - i - 1; j++) {
      if (arr[j] > arr[j + 1]) {
        temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
        swapped = 1;
      }
    }
    if (!swapped) {
      break;
    }
  }
}
```

```
int main() {
  int matrix[ROWS][COLS] = {
    {10, 20, 30, 40, 50, 60, 70, 80, 90, 100},
    {110, 120, 130, 140, 150, 160, 170, 180, 190, 200},
    {210, 220, 230, 240, 250, 260, 270, 280, 290, 300},
    {310, 320, 330, 340, 350, 360, 370, 380, 390, 400},
    {410, 420, 430, 440, 450, 460, 470, 480, 490, 500},
    {510, 520, 530, 540, 550, 560, 570, 580, 590, 600},
    {610, 620, 630, 640, 650, 660, 670, 680, 690, 700},
  };

  printf("Original Matrix:\n");
  for (int i = 0; i < ROWS; i++) {
    for (int j = 0; j < COLS; j++) {
      printf("%3d ", matrix[i][j]);
    }
    printf("\n");
  }

  int oddElements[(ROWS + 1) / 2];
  int k = 0;

  for (int i = 1; i < ROWS; i += 2) {
    oddElements[k++] = matrix[i][COLS - 1];
  }
```

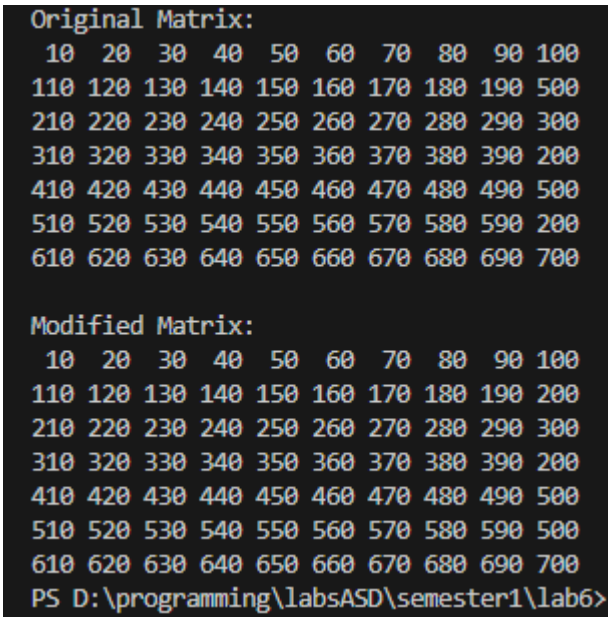
```
  bubbleSortWithFlag(oddElements, k);

  k = 0;
  for (int i = 1; i < ROWS; i += 2) {
    matrix[i][COLS - 1] = oddElements[k++];
  }

  printf("\nModified Matrix:\n");
  for (int i = 0; i < ROWS; i++) {
    for (int j = 0; j < COLS; j++) {
      printf("%3d ", matrix[i][j]);
    }
    printf("\n");
  }

  return 0;
}
```

Тестування програми

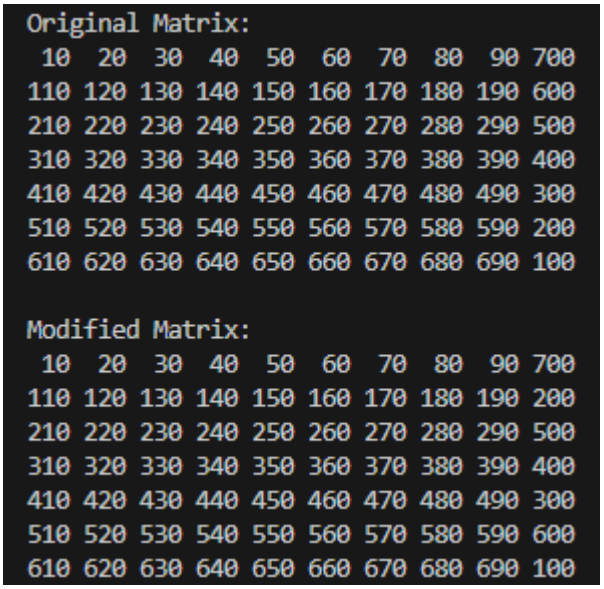
1) Для вже відсортованої матриці (тобто, яка має “правильний” порядок) програма видає такий результат:



2) Для невідсортованої матриці матриці програма повертає наступний результат:



3) Для обернено відсортованої матриці програма повертає наступний результат:



Висновок

В лабораторній роботі №2.2 я теоретично засвоїв різні алгоритими сортування, та за допомогою одного з них виконав поставлене завдання з сортування елементів у двовимірному масиві. Програма працює коректно.