Ministry of Education and Science of Ukraine

National Technical University of Ukraine

«Kyiv Polytechnic Institute. Igor Sikorsky »

Faculty of Informatics and Computer Technologies

Department of Computer Engineering

LAB № 3

from the discipline "Theory of Algorithms"

on the topic «Quick sort method»

PERFORMED BY:

1st year student

group ІП-93

Dominskyi V.O.

The credit - 9311

Variant – 11

CHECKED:

Associate Professor of OT

c.t.s.,s.r.

Antoniuk А.І.

Kyiv - 2020

**TASK**

**Goal:**

implementation of three modifications of the Quick Sort algorithm and comparison of their performance.

**Option task:**

Implement three modifications of the Quick Sort algorithm and compare their performance. The speed of the algorithms is compared on the basis of counting the number of comparisons of the array elements during the operation of the algorithms.

**CODE**

**using** System**;**

class Script

**{**

**public** static int Swap**=**0**;**

**public** static int Swap1**=**0**;**

**public** static int Swap2**=**0**;**

**public** static void Main**()**

**{**

int **[]**Array **=** **{**10**,** 7**,** 8**,** 9**,** 1**,** 5**};**

int **[]**Array1 **=** **{**7**,** 8**,** 1**,** 9**,** 10**,** 5**};**

int **[]**Array2 **=** **{**5**,** 7**,** 9**,** 8**,** 1**,** 10**};**

int Numbers **=** Array**.**Length**;**

Console**.**WriteLine**(**"Arrays are : First = "**);**

PrintArray**(**Array**,** Numbers**);**

Console**.**WriteLine**(**"second = "**);**

PrintArray**(**Array1**,** Numbers**);**

Console**.**WriteLine**(**"and third = "**);**

PrintArray**(**Array2**,** Numbers**);**

QuickSort**(**Array**,** 0**,** Numbers**-**1**);**

QuickSort1**(**Array1**,** 0**,** Numbers**-**1**);**

QuickSort2**(**Array2**,** 0**,** Numbers**-**1**);**

Console**.**WriteLine**(**"Final array = "**);**

PrintArray**(**Array**,** Numbers**);**

Console**.**WriteLine**(**"Final array1 = "**);**

PrintArray**(**Array1**,** Numbers**);**

Console**.**WriteLine**(**"Final array2 = "**);**

PrintArray**(**Array2**,** Numbers**);**

**}**

static int Split**(**int **[]**Array**,** int Left**,** int Right**)**

**{**

int CentralPoint **=** Array**[**Right**];**

int i **=** **(**Left **-** 1**);**

**for** **(**int j **=** Left**;** j **<** Right**;** j**++)**

**{**

Swap**++;**

Console**.**WriteLine**(**"Swaps = " **+** Swap**);**

**if** **(**Array**[**j**]** **<** CentralPoint**)**

**{**

i**++;**

int NotPermanent **=** Array**[**i**];**

Array**[**i**]** **=** Array**[**j**];**

Array**[**j**]** **=** NotPermanent**;**

**}**

**}**

int NotPermanent\_1 **=** Array**[**i**+**1**];**

Array**[**i**+**1**]** **=** Array**[**Right**];**

Array**[**Right**]** **=** NotPermanent\_1**;**

**return** i**+**1**;**

**}**

static int Split1**(**int **[]**Array**,** int Left**,** int Right**)**

**{**

int CentralPoint **=** Array**[**Right**];**

int i **=** **(**Left **-** 1**);**

**for** **(**int j **=** Left**;** j **<** Right**;** j**++)**

**{**

Swap1**++;**

Console**.**WriteLine**(**"Swaps1 = " **+** Swap1**);**

**if** **(**Array**[**j**]** **<** CentralPoint**)**

**{**

i**++;**

int NotPermanent **=** Array**[**i**];**

Array**[**i**]** **=** Array**[**j**];**

Array**[**j**]** **=** NotPermanent**;**

**}**

**}**

int NotPermanent\_1 **=** Array**[**i**+**1**];**

Array**[**i**+**1**]** **=** Array**[**Right**];**

Array**[**Right**]** **=** NotPermanent\_1**;**

**return** i**+**1**;**

**}**

static int Split2**(**int **[]**Array**,** int Left**,** int Right**)**

**{**

int CentralPoint **=** Array**[**Right**];**

int i **=** **(**Left **-** 1**);**

**for** **(**int j **=** Left**;** j **<** Right**;** j**++)**

**{**

Swap2**++;**

Console**.**WriteLine**(**"Swaps2 = " **+** Swap2**);**

**if** **(**Array**[**j**]** **<** CentralPoint**)**

**{**

i**++;**

int NotPermanent **=** Array**[**i**];**

Array**[**i**]** **=** Array**[**j**];**

Array**[**j**]** **=** NotPermanent**;**

**}**

**}**

int NotPermanent\_1 **=** Array**[**i**+**1**];**

Array**[**i**+**1**]** **=** Array**[**Right**];**

Array**[**Right**]** **=** NotPermanent\_1**;**

**return** i**+**1**;**

**}**

static void QuickSort**(**int **[]**Array**,** int Left**,** int Right**)**

**{**

**if** **(**Left **<** Right**)**

**{**

int SomeNumber **=** Split**(**Array**,** Left**,** Right**);**

QuickSort**(**Array**,** Left**,** SomeNumber**-**1**);**

QuickSort**(**Array**,** SomeNumber**+**1**,** Right**);**

**}**

**}**

static void QuickSort1**(**int **[]**Array**,** int Left**,** int Right**)**

**{**

**if** **(**Left **<** Right**)**

**{**

int SomeNumber **=** Split1**(**Array**,** Left**,** Right**);**

QuickSort1**(**Array**,** Left**,** SomeNumber**-**1**);**

QuickSort1**(**Array**,** SomeNumber**+**1**,** Right**);**

**}**

**}**

static void QuickSort2**(**int **[]**Array**,** int Left**,** int Right**)**

**{**

**if** **(**Left **<** Right**)**

**{**

int SomeNumber **=** Split2**(**Array**,** Left**,** Right**);**

QuickSort2**(**Array**,** Left**,** SomeNumber**-**1**);**

QuickSort2**(**Array**,** SomeNumber**+**1**,** Right**);**

**}**

**}**

static void PrintArray**(**int **[]**Array**,** int Numbers**)**

**{**

**for** **(**int i **=** 0**;** i **<** Numbers**;** **++**i**)**

**{**

Console**.**Write**(**Array**[**i**]** **+** " "**);**

**}**

Console**.**WriteLine**();**

**}**

**}**

**RESULTS OF THE PROGRAM WORK**

The input arrays are = [10, 7, 8, 9, 1, 5], [7, 8, 1, 9, 10, 5], [5, 7, 9, 8, 1, 10].

Output array: = [1 5 7 8 9 10]. Swaps = 11,9 and 15

**CONCLUSIONS**

I got acquainted with the topic of laboratory work.

Have acquired relevant work skills.

An appropriate test program has been developed.

The running time of the sorting algorithm depends on the balance characterizing the partition. Balance, in turn, depends on which element is selected as the reference (relative to which element is being partitioned)