

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра Информационная безопасность вычислительных систем и сетей

Методы сортировки
(наименование работы)

ОТЧЕТ

по лабораторной работе №5

по дисциплине

Технологии программирования
(наименование дисциплины)

РУКОВОДИТЕЛЬ:

(подпись)

Капранов С.Н.
(фамилия, и., о.)

СТУДЕНТ:

(подпись)

Савельев М.А.
(фамилия, и., о.)

18-ИСТ-4
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород, 2020

Содержание

Задача	3
Основная часть отчета.....	4
Листинг программы.....	4
Входные и выходные данные	5

Задача

16 вариант: Реализовать «быструю» сортировку

Основная часть отчета

Программа написана на языке C++ в среде разработки VisualStudio 2017.

Листинг программы

Quick_sort.cpp

```
*****
#include <iostream>
#include "quick_sort.h"

using namespace std;

//--- Вывод на экран ---\\
void print(int arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << "|";
    }
    cout << endl;
}
//--- ---\\

int main()
{
    int n; // размер массива
    cout << "Array Size: ";
    //--- Обработка ввода ---\\
    while (!(cin >> n) || n < 0)
    {
        cout << "Uncorrect size, try again \n";
        cout << "Array Size: ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    cout << endl;
    //--- ---\\

    int* arr = new int[n]; // массив размером n
    //--- Ввод n элементов массив поочередно ---\\
    for (int i = 0; i < n; i++)
    {
        cout << "Array[" << i + 1 << "]: ";
        while (!(cin >> arr[i]))
        {
            cout << "Uncorrect number, try again \n";
            cout << "Array[" << i + 1 << "]: ";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
        cout << endl;
    }
    //--- ---\\

    print(arr, n); // вывод на экран введенной последовательности
    quickSort(arr, 0, n - 1); // функция сортировки
    print(arr, n); // вывод на экран результата сортировки
    system("pause");
}
```

Quick sort.h

```
*****
#include <iostream>

using namespace std;

//--- Функция "быстрой" сортировки ---\\
void quickSort(int arr[], int left, int right)
{
    int i = left, j = right; // границы массива
    int tmp; // переменная, временно хранящая в себе значения
    int pivot = arr[(left + right) / 2]; // нахождение середины массива

    while (i <= j) { // пока не середина
        while (arr[i] < pivot) // пока значения слева меньше срединного
            i++;
        while (arr[j] > pivot) // пока значения справа больше срединного
            j--;
        if (i <= j)
        {
            //--- Сwap значений ---\\
            tmp = arr[i];
            arr[i] = arr[j];
            arr[j] = tmp;
            //--- ---\\
            i++;
            j--;
        }
    }
    if (left < j)
        quickSort(arr, left, j); // вызов функции для левой части массива
    if (i < right)
        quickSort(arr, i, right); // вызов функции для правой части массива
}
//--- ---\\
```

Входные и выходные данные

```
Array Size: error
Uncorrect size, try again
Array Size: -78
Uncorrect size, try again
Array Size: 5

Array[1]: *hfkajs
Uncorrect number, try again
Array[1]: nice
Uncorrect number, try again
Array[1]: 89

Array[2]: -745

Array[3]: 321

Array[4]: -7468

Array[5]: 456
```

```
Unsort Array:
89 | -745 | 321 | -7468 | 456 |
Sort Array:
-7468 | -745 | 89 | 321 | 456 |
```

Рис. 1. Входные данные

Рис. 2. Выходные данные