

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Алгоритмы и структуры данных»
Тема: Сортировка

Студент гр. 9384

Прашутинский К.И.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Разработать сортировку слиянием.

Задание. (Вариант № 14)

Сортировка массивов слиянием – простое слияние, рекурсивная реализация.

Выполнение работы.

Были разработаны функции void Merge() и void MergeSort(), которые рекурсивно разделяют и сортируют целочисленный массив.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	3 1 5 3	1 3 5	
2.	100 2 77 62 32 44 56 100 53 47 20 26 83 70 37 7 7 98 65 85 49 34 55 27 38 29 10 30 56 68 70 10 69 47 72 1 90 27 100 43 73 19 69 56 89 5 63 95 2 27 80 51 61 34 77 98 63 86 28 18 54 98 27 22 44 98 23 34 25 22 76 97 41 44 53 29 49 15 24 51 42 4 1 3 37 78 1 99 64 28 17 17 26 43 39 70 41 61 3 65 83	1 1 1 2 2 3 3 4 5 7 7 10 10 15 17 17 18 19 20 22 22 23 24 25 26 26 27 27 27 27 28 28 29 29 30 32 34 34 34 37 37 38 39 41 41 42 43 43 44 44 44 47 47 49 49 51 51 53 53 54 55 56 56 56 61 61 62 63 63 64 65 65 68 69 69 70 70 70 72 73 76 77 77 78 80 83 83 85 86 89 90 95 97 98 98 98 98 99 100 100	

Выводы.

При выполнении работы было подкреплено понятие рекурсия, была реализована сортировка слиянием.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <string>
```

```
#include "MergeSort.h"
```

```
void main()
```

```
{
```

```
    setlocale(LC_ALL, "Rus");
```

```
    std::ifstream input;
```

```
    std::string path;
```

```
    int c = 0;
```

```
    int n;
```

```
    int *A;
```

```
    std::cout << "Выберите считывание:\n\t 1)из консоли\n\t 2)из  
файла\n";
```

```
    while (!c) {
```

```
        std::cout << ">>>";
```

```
        std::cin >> c;
```

```
        switch (c)
```

```
        {
```

```
        case 1:
```

```
            std::cout << "Размер массива > ";
```

```
            std::cin >> n;
```

```
            A = new int[n];
```

```

for (int i = 0; i < n; i++)
{
    std::cout << i << " элемент > ";
    std::cin >> A[i];
}
MergeSort(A, 0, n - 1);
std::cout << "Упорядоченный массив: ";
for (int i = 0; i < n; i++)
    std::cout << A[i] << " ";
std::cout << "\n";
delete[]A;
break;

```

case 2:

```

std::cout << "Введите путь до файла: ";
std::cin >> path;
input.open(path);
if (input.is_open()) {
    input >> n;
    A = new int[n];
    for (int i = 0; i < n; i++)
    {
        input >> A[i];
    }
    MergeSort(A, 0, n - 1);
    std::cout << "Упорядоченный массив: ";
    for (int i = 0; i < n; i++)
        std::cout << A[i] << " ";
    std::cout << "\n";
    delete[]A;
}

```

```

        }
        else {
            std::cout << "Файл не открыт!\n";
        }
        input.close();
        break;
    default:
        c = 0;
        break;
    }
}
system("pause");
}

```

Название файла: MergeSort.cpp

```
#include "MergeSort.h"
```

```
void Merge(int *A, int first, int last)
```

```

{
    int middle = 0, start = 0, final = 0;
    int n = last - first + 1;
    int *mas = new int[n];
    middle = (first + last) / 2;
    start = first;
    final = middle + 1;
    for (int j = first; j <= last; j++)
        if ((start <= middle) && ((final > last) || (A[start] < A[final])))
        {
            mas[j - first] = A[start];
            start++;
        }
    }

```

```

        }
    else
    {
        mas[j - first] = A[final];
        final++;
    }

    for (int j = first; j <= last; j++) A[j] = mas[j - first];
    delete[] mas;
}

void MergeSort(int *A, int first, int last)
{
    if (first < last)
    {
        MergeSort(A, first, (first + last) / 2);
        MergeSort(A, (first + last) / 2 + 1, last);
        Merge(A, first, last);
    }
}

```

Название файла: MergeSort.h

```
#ifndef MERGESORT_H
```

```
#define MERGESORT_H
```

```
void Merge(int *A, int first, int last);
```

```
void MergeSort(int *A, int first, int last);
```

```
#endif // !MERGESORT_H
```