

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01 – «Информатика и вычислительная
техника»

**Отчет по
лабораторной работе 15**

Выполнил студент гр. ИВТ-23-16
Давыдов Андрей Юрьевич

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка)

(подпись)

(дата)

г. Пермь, 2023

Постановка задачи

Многофазная и естественная сортировка

Код программы:

```
#include <iostream>
#include <fstream>

using namespace std;

void merge(int arr[], int l, int m, int r) {
    int n1 = m - l + 1;
    int n2 = r - m;

    int* L = new int[n1];
    int* R = new int[n2];

    for (int i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    int i = 0, j = 0, k = l;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
        else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }

    delete[] L;
    delete[] R;
}
```

```

void naturalMergeSort(int arr[], int n) {
    bool isSorted = false;
    while (!isSorted) {
        isSorted = true;
        int l = 0;

        while (l < n) {
            int m = l;
            while (m < n - 1 && arr[m] <= arr[m + 1]) {
                m++;
            }

            if (m == n - 1) break;

            int r = m + 1;
            while (r < n - 1 && arr[r] <= arr[r + 1]) {
                r++;
            }

            merge(arr, l, m, r);

            if (r < n - 1) {
                isSorted = false;
            }

            l = r + 1;
        }
    }
}

void multiPhaseSort(int arr[], int n) {
    for (int size = 1; size < n; size *= 2) {
        for (int left = 0; left < n; left += 2 * size) {
            int mid = min(left + size - 1, n - 1);
            int right = min(left + 2 * size - 1, n - 1);

            merge(arr, left, mid, right);
        }
    }
}

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

```

```

int main() {
    system("chcp 1251 > Null");
    int arr[] = { 12, 4, 5, 6, 7, 3, 1, 15 };
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Исходный массив: ";
    printArray(arr, n);

    int choice;
    cout << "Выберите метод сортировки:\n";
    cout << "1. Многофазная сортировка\n";
    cout << "2. Естественная сортировка\n";
    cout << "Введите номер выбора: ";
    cin >> choice;

    switch (choice) {
        case 1:
            multiPhaseSort(arr, n);
            break;
        case 2:
            naturalMergeSort(arr, n);
            break;
        default:
            cout << "Неверный выбор. Пожалуйста, введите 1 или 2.\n";
            return 1;
    }

    cout << "Отсортированный массив: ";
    printArray(arr, n);

    return 0;
}

```

Тест программы

```

Исходный массив: 12 4 5 6 7 3 1 15
Выберите метод сортировки:
1. Многофазная сортировка
2. Естественная сортировка
Введите номер выбора: 1
Отсортированный массив: 1 3 4 5 6 7 12 15

```

```

Исходный массив: 12 4 5 6 7 3 1 15
Выберите метод сортировки:
1. Многофазная сортировка
2. Естественная сортировка
Введите номер выбора: 2
Отсортированный массив: 1 3 4 5 6 7 12 15

```