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: Сортировки

Студент гр. 9304 _____ Попов Д.С.

Преподаватель _____ Филатов А.Ю

Санкт-Петербург

2020

Цель работы.

Изучить базовые методы сортировки, реализовать пользовательскую сортировку на языке C++.

Задание.

20) Поразрядная сортировка.

Выполнение работы.

Программа принимает из стандартного потока ввода строку и конвертирует ее в вектор значений, после чего передается функции radix для сортировки. Поразрядная сортировка подразумевает перебор элементов в зависимости от того, какая цифра находится в определённом разряде. После обработки всех разрядов массив оказывается упорядоченным. В данной программе используется один из вариантов поразрядной сортировки, а именно LSD — сортировка по младшим разрядам. Элементы перебираются по порядку и группируются по самому младшему разряду (сначала все, заканчивающиеся на 0, затем заканчивающиеся на 1, ..., заканчивающиеся на 9). Возникает новая последовательность. Затем группируются по следующему разряду с конца, затем по следующему и т.д. пока не будут перебраны все разряды, от младших к старшим.

Разработанный программный код см. в приложении А.

Формат входных и выходных данных.

На вход программе должен подаваться набор беззнаковых целочисленных значений разделенных пробелом, отрицательные числа отбрасываются, а иные символы приводят к досрочному завершению программы. На выходе получаем упорядоченный массив значений.

Тестирование.

Для проведения тестирования был написан bash-скрипт ./script .Скрипт запускает программу где в качестве входных аргументов служат заранее подготовленные файлы, расположенные в папке ./Tests

```
Test 3:
Начальная строка = 2456 1234 6545 7567 2345 9431 2341
Итерация 1:
Блок 1: [9431, 2341]
Блок 4: [1234]
Блок 5: [6545, 2345]
Блок 6: [2456]
Блок 7: [7567]
Объединение блоков: 9431 2341 1234 6545 2345 2456 7567
Итерация 2:
Блок 3: [9431, 1234]
Блок 4: [2341, 6545, 2345]
Блок 5: [2456]
Блок 6: [7567]
Объединение блоков: 9431 1234 2341 6545 2345 2456 7567
Итерация 3:
Блок 2: [1234]
Блок 3: [2341, 2345]
Блок 4: [9431, 2456]
Блок 5: [6545, 7567]
Объединение блоков: 1234 2341 2345 9431 2456 6545 7567
Итерация 4:
Блок 1: [1234]
Блок 2: [2341, 2345, 2456]
Блок 6: [6545]
Блок 7: [7567]
Блок 9: [9431]
Объединение блоков: 1234 2341 2345 2456 6545 7567 9431
Результат: 1234 2341 2345 2456 6545 7567 9431
```

Рисунок 1 — Часть вывода скрипта.

Выводы.

Была изучена поразрядная сортировка и реализована программа с ее использованием. Сложность данной сортировки составлет $O(n \times k)$, где n — количество элементов, а k — максимальное количество разрядов.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <vector>
#include <cmath>
#include <iterator>
#include <sstream>


void radix(std::vector<int>& digitArr);     // Поразрядная сортировка


int main(){

    std::string inputString{};
    getline(std::cin, inputString);

    std::stringstream strStream(inputString);
    std::vector<int> vec {};

    std::copy(std::istream_iterator<int>(strStream), {},
back_inserter(vec));

    radix(vec);

    std::cout << "Результат: ";
    for(const auto& digit : vec){
        std::cout << digit << ' ';
    }
    std::cout << std::endl;

    return 0;
}



void radix(std::vector<int>& digitArr){

    std::vector<std::vector<int>> vec {};   // Для каждого разряда свой
блок
    unsigned deegre = 0;                    // Разряд числа
    unsigned level;                         // Позиция блока в массиве
    bool checkRank = 0;                     // Проверка на наличие
отличных от 0 рангов
    vec.resize(10);

    while(1){

        // Разбрасываем по соответствующим блокам
        deegre++;
        for(size_t i = 0; i < digitArr.size(); i++){
            if(digitArr[i] >= 0){
```

```
                level = digitArr[i] / ((int)pow(10, deegre - 1)) % 10;
                if(level){
                    checkRank = 1;
                }
                vec[level].emplace_back(digitArr[i]);
            }
        }

        // Перезапись базового вектора
        digitArr.clear();
        for(size_t i = 0; i < vec.size(); i++){
            for(size_t j = 0; j < vec[i].size(); j++){
                digitArr.emplace_back(vec[i][j]);
            }
        }

        // Если все элементы в 0 блоке - ливаем
        if(!checkRank){
            break;
        }else{
            checkRank = 0;
        }

        // Вывод промежуточного результата
        std::cout << "Итерация " << deegre << ":\n";
        for(size_t i = 0; i < vec.size(); i++){
            if(vec[i].size()){
                std::cout << "Блок " << i << ": [";
                for(size_t j = 0; j < vec[i].size(); j++){
                    std::cout << vec[i][j];
                    if(!(j == vec[i].size() - 1)){
                        std::cout << ", ";
                    }
                }
                std::cout << "]\n";
            }
        }
        std::cout << "Объединение блоков: ";
        for(const auto& digit : digitArr){
            std::cout << digit << ' ';
        }
        std::cout << "\n";

        // Чистим для следующей итерации
        for(size_t i = 0; i < 10; i++){
            vec[i].clear();
        }

    }
}
```

Название файла: script

```
#!/bin/bash
for n in {1..7}
do
      arg=$(cat Tests/test$n.txt)
      echo -e "\nTest $n:"
      echo "Начальная строка = $arg"
      ./lab4 < Tests/test$n.txt
done
```