

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

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

Студентка гр. 9303

Булыно Д. А.

Преподаватель

Филатов А. Ю.

Санкт-Петербург

2020

Цель работы.

Формирование практических навыков программирования сортировки на языке программирования C++ путём решения поставленной задачи.

Основные теоретические положения. Описание алгоритма.

Поразрядная сортировка — алгоритм сортировки, который выполняется за линейное время.

Исходно предназначен для сортировки целых чисел, записанных цифрами. Но так как в памяти компьютеров любая информация записывается целыми числами, алгоритм пригоден для сортировки любых объектов, запись которых можно поделить на «разряды», содержащие сравнимые значения. Например, так сортировать можно не только числа, записанные в виде набора цифр, но и строки, являющиеся набором символов, и вообще произвольные значения в памяти, представленные в виде набора байт.

Сравнение производится поразрядно: сначала сравниваются значения одного крайнего разряда, и элементы группируются по результатам этого сравнения, затем сравниваются значения следующего разряда, соседнего, и элементы либо упорядочиваются по результатам сравнения значений этого разряда внутри образованных на предыдущем проходе групп, либо переупорядочиваются в целом, но сохраняя относительный порядок, достигнутый при предыдущей сортировке. Затем аналогично делается для следующего разряда, и так до конца.

Накапливать при каждом проходе сведения о группах можно разными способами — например в списках, в деревьях, в массивах, выписывая в них либо сами элементы, либо их индексы и т. п.

Существует нестабильный вариант рекурсивной побитовой сортировки, выполняющейся непосредственно в сортируемом массиве: на первом проходе движение идёт навстречу друг другу, в начале массива ищется элемент с 1 в первом битовом разряде, в конце массива ищется элемент с 0 в том же разряде. Найденные элементы меняются местами, и так до тех пор, пока рассматриваемые индексы не встретятся. Таким образом в начале массива, до места встречи

индексов, собираются все элементы с битом равным 0, а после этого индекса — все элементы с равным 1. Далее рекурсивно можно полностью аналогично перебрать получившиеся поддиапазоны массива, сравнивая значения второго и последующих разрядов, и переставляя элементы местами.

Сложность по времени в лучшем случае $O(n)$, в худшем - $O(k * n)$.

Сложность о памяти $O(n + k)$.

Задание.

Вариант 20

Поразрядная сортировка.

Выполнение работы.

Для выполнения программы был написан класс `stack` и функции `int pow(int a, int b)`, `int* foo(int *Arr, int count)`. Методы класса предназначены для временного хранения, обработки и записи промежуточных значений работы сортировки. С помощью функции `int* foo(int *Arr, int count)` значения массива сортируются поразрядно. Для этого была написана функция `int pow(int a, int b)`, благодаря которой считаются разряды элементов исходного массива.

Код программы см. в приложении А.

Результаты работы программы см. в файле «result.txt».

Выводы.

В ходе выполнения лабораторной работы была изучена поразрядная сортировка. В результате выполнения лабораторной работы была написана программа, которая сортирует массив с помощью поразрядной сортировки.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
using namespace std;

int pow( int a, int b ){
    int c = 1;
    for (int i = 0; i < b; i++){
        c*=a;
    }
    return c;
}

class stack{
private:
    int count;
    int **Arr;
    void getSize(){
        count++;
        int **tmp = new int*[count];

        for (int i = 0; i < count; i++){
            tmp[i] = new int[10];
            for (int j = 0; j < 10; j++){
                tmp[i][j] = -1;
            }
        }
        for (int i = 0; i < count-1; i++){
            tmp[i] = Arr[i];
        }
        Arr = tmp;
    }
public:
    stack(){
        count = 0;
        getSize();
    }
    void push(int val, int index){
        if(!count) getSize();
        bool flag = true;
        for (int i = 0; i < count; i++){
            if(!Arr[i][index]){
```

```

        Arr[i][index] = val;
        false;
    }
}
if(flag){
    getSize();
    Arr[count-1][index] = val;
}
};

int pop(){
    int flag = true;
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < count; j++){
            if(Arr[j][i] != -1){
                int tmp = Arr[j][i];
                Arr[j][i] = -1;
                false;
                return tmp;
            }
        }
    }
    if (flag){
        for (int i = 0; i < count; i++){
            delete [] Arr[i];
        }
        count = 0;
    }
    return -1;
}

void print(){
    for (int i = 0; i < 10; i++){
        cout << "[" << i << "]: ";
        for (int j = 0; j < count; j++){
            if(Arr[j][i] != -1){
                cout << "[ "<< Arr[j][i] << " ] | ";
            }
        }
        cout << endl;
    }
}

~stack(){
    if(count){
        for (int i = 0; i < count; i++){

```

```

        delete [] Arr[i];
    }
    count = 0;
}
};

```

```

int* foo(int *Arr, int count){
    int max = Arr[0];
    for (int i = 0; i < count; i++){
        if (Arr[i] > max) max = Arr[i];
    }

    stack st;
    int Radix = 1, c;

    while (max >= pow(10, Radix-1)){
        c = pow(10, Radix);

        for (int i = 0; i < count; i++){
            cout << "Разряд: " << Radix << "\n";
            st.push(Arr[i], (Arr[i]%c-Arr[i]%(c/10))/pow(10, Radix-1));
            st.print();
            cout << endl;
        }

        for (int i = 0; i < count; i++){
            cout << "Разряд: " << Radix << "\n";
            Arr[i] = st.pop();
            st.print();
            cout << endl;
        }

        Radix++;
    }
    return Arr;
}

int main(){
    int n;
    cout << "Введите количество элементов массива: n = ";
    cin >> n;
}

```

```

    if (!n){
        cout << "Размер массива не задан" << endl;
        return 0;
    }
    int mass[n];
    cout << "Введите массив: a[n] = ";
    for (int i = 0; i < n; i++){
        cin >> mass[i];
    }
    int *arr = new int[n];
    for (int i = 0; i < n; i++){
        arr[i] = mass[i];
    }

    arr = foo(arr, n);

    cout << "Результат поразрядной сортировки:" << endl;
    for (int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    return 0;
}

```