

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: Алгоритмы сортировки

Студент гр. 9382

Дерюгин Д.А.

Преподаватель

ФирсовМ.А.

Санкт-Петербург

2020

Цель работы.

Изучить принципы сортировки массива. Создать программу, сортирующую заданных массив чисел циклической сортировкой.

Основные теоретические положения.

Алгоритм сортировки - алгоритм, который упорядочивает элементы в массиве. Все члены входного массива должны быть переставлены так, что $a_i \leq a_{i+1}$, для любых i от 0 до n

Задание.

Вариант 7 Циклическая сортировка.

Описание алгоритма.

На входе есть неотсортированный массив чисел. Запускается цикл, который проходит по всему массиву. Пусть текущий элемент массива - q . Каждый элемент q массива сравнивается со всеми элементами этого же массива, которые стоят после q . Если какое то число, которое стоит дальше q и оно меньше его, то q сдвигается на 1 позицию вперед. В итоге q сдвигается на позицию, равную $i+k$ (где i - начальная позиция данного элемента, а k - количество чисел, которые меньше q и лежат дальше по массиву). Число, которое будет лежать на новом месте q теперь будет являться q . Это происходит до того, пока на позиции самого первого элемента q не будет стоять уже конечный элемент. Если это произошло, то внутренний цикл заканчивается и внешний цикл сдвигается на 1 вперед по массиву. Теперь данный элемент будет q . Так происходит, пока внешний цикл не пройдет весь массив.

Чтобы избежать бесконечного цикла, который может произойти в результате того, что есть несколько одинаковых чисел, можно сделать условие, которое проверяет: является ли число, на месте которого будет стоять q равным этому q . Если это так, то q помещаем на 1 позицию дальше. (так происходит до тех пор, пока на позиции не будет числа, которое отличается от q).

Среднее время: $O(n^2/2)$

Если массив состоит из повторяющихся чисел, то циклическая сортировка справится в десятки раз быстрее, чем обычная сортировка выбором,

однако, если нет повторяющихся чисел, цикличная сортировка работает даже в несколько раз медленнее, чем обычная сортировка выбором, так как приходится больше бегать по массиву и чаще сравнивать.

Описание функций и структур данных.

`void cycleSort(int lengthOfArray, int *arr)` - функция, которая выполняет бинго-сортировку.

`int lengthOfArray` - длина массива.

`int *arr` - указатель на сам массив.

`enterFromFile()` - считывает данные из файла.

`createArray()` - создает массив.

Тестирование.

Результаты тестирования представлены в табл. 1

Табл. 1 - результаты тестирования

Входные данные	Выходные данные	Комментарий
5 -3 6 0 4	<p>Current value 5. Current position: 0 New position of this value is 3</p> <p>Current value 0. Current position: 3 New position of this value is 1</p> <p>Current value -3. Current position: 1 New position of this value is 0</p> <p>Circle is (5, 0, -3, 5)</p> <p>Current value 0.</p>	Полный вывод. В следующих тестах будет выводиться просто результат

	<p>Current position: 1</p> <p>This value on the position</p> <p>Current value 6.</p> <p>Current position: 2</p> <p>New position of this value is 4</p> <p>Current value 4.</p> <p>Current position: 4</p> <p>New position of this value is 2</p> <p>Circle is (6, 4, 6)</p> <p>Current value 5.</p> <p>Current position: 3</p> <p>This value on the position</p> <p>Current value 6.</p> <p>Current position: 4</p> <p>This value on the position</p> <p>-3 0 4 5 6</p>	
-2 -91 44 20 -100 -2 99 41 7 52 1	-100 -91 -2 -2 1 7 20 41 44 52 99	Работает с отрицательными числами.

0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	Все нули обрабатывает
---------------------	---------------------	-----------------------

25 -95 98 -71 11 67	-95 -71 11 25 67 98	
-453 5346 -4364 346 -346 -346 -346 634 -754 4 0 0 -347 345 478 -345 -754	-4364 -754 -754 -453 -347 - 346 -346 -346 -345 0 0 4 345 346 478 634 5346	
1	1	

Выводы.

Были изучены принципы сортировки массивов. Создали программу, которая при помощи циклической сортировки сортирует массив.

ПРИЛОЖЕНИЕ А.

ИСХОДНЫЙ КОД ПРОГРАММЫ.

```
#include <iostream>
#include <fstream>
using namespace std;
template <typename T>
void cycleSort(int lengthOfArray, T *arr) {
    string circle = "(";
    int value, pos, extra; // value - buffer value; pos - current position; extra
- extra variable for swap
    for (int cycleStart = 0; cycleStart < lengthOfArray; cycleStart++) {
        value = arr[cycleStart];
        pos = cycleStart;
        cout<<"Current value "<<value<<". Current position: "<<pos<<endl;
        for (int i = cycleStart + 1; i < lengthOfArray; i++) // search for number
which less then value
            if (arr[i] < value) pos+= 1;
        if (pos == cycleStart) { // if current value on the position
            cout<<"This value on the position\n";
            continue;
        }
        while (value == arr[pos]) pos+= 1; // continue if value == next position
        cout<<"New position of this value is "<<pos<<endl;
        // swap variables
        circle+= to_string(value) + ", ";
        extra = arr[pos];
        circle+=to_string(extra) + ", ";
        arr[pos] = value;
        value = extra;
        //start circle
        while (pos != cycleStart) {
            cout<<"Current value "<<value<<". Current position: "<<pos<<endl;
            pos = cycleStart;
            for (int i = cycleStart + 1; i < lengthOfArray; i++) // search for
number which less then value
                if (arr[i] < value) pos+= 1;
            while (value == arr[pos]) pos+= 1; // continue if value == next
position
            cout<<"New position of this value is "<<pos<<endl;
            //swap variables
            extra = arr[pos];
            arr[pos] = value;
            value = extra;
            circle+= to_string(extra);
            if (pos != cycleStart) circle+= ", ";
        }
    }
}
```

```

        cout<<"Circle is "<<circle<<")"<<endl<<endl;
        circle = "(";
    }

    for (int i = 0 ; i < lengthOfArray; i ++) {
        cout<<arr[i]<<" ";
    }
}

template <typename T>
void enterFromFile() {
    string path = "input.txt";// path to input file
    int index = 0;
    int array[10000];
    //open file
    ifstream fin;
    fin.open(path);
    //if cannot open file
    if (!fin.is_open()) {
        cout<<"Cannot open file";
        exit(1);
    }
    //reading file line by line
    while(!fin.eof()) {
        fin>>array[index];
        index++;
    }
    fin.close();//close file
    cycleSort(index, array);
}

template <typename T>
void createArray() {
    string lengthOfArray;// length of array
    int index = 0;// index of array
    bool checkString;
    int typeOfInput;// 1 if console
    while (true) {
        cout<<"\nIf you want to end program press '0'\n\"Enter '1' if you wanna
write down array in console otherwise write down any letter or number:\n";
        cin>>typeOfInput;
        //input from console
        if (typeOfInput == 0) break;
        if (typeOfInput == 1) {
            do {while length<0
                checkString = true;
                cout<<"\nEnter length of array"<<endl;

                cin>>lengthOfArray;
                for (int i = 0 ; i < lengthOfArray.length(); i++) {
                    if (lengthOfArray[i] <='0' || lengthOfArray[i] > '9') {
                        if (lengthOfArray[i] == '0' && i > 0) break;
                        checkString = false;
                        break;
                    }
                }
            } while (!checkString);
            T array[stoi(lengthOfArray)];
            cout<<"enter array using tab:\n";
            for (int i = 0; i < stoi(lengthOfArray); i++)
                cin>>array[i];

            cycleSort(stoi(lengthOfArray), array);//sort func

```

```
    }
    else {
        enterFromFile<int>();
    }

}

}

int main() {
    createArray<int>();
    return 0;
}
```