

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

Гурин С.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить способ реализации алгоритмов сортировки на языке C++

Задание.

ВАРИАНТ 3.

Двухпутевая сортировка бинарными вставками(при каждой вставке перемещаются не более половины элементов отсортированной последовательности).

Анализ задачи.

Задача заключается в том, что нужно реализовать функции сортировки массива чисел.

Выполнение работы.

При запуске программы пользователь должен ввести в input.txt последовательность чисел, которую необходимо отсортировать. Если файл не оказался пустым, то пользователю выводится отсортированный массив, в противном случае, если файл пустой, пользователю выводится сообщение о том, что файл пустой и программа завершает свою работу.

Алгоритм сортировки происходит с помощью двухпутевых вставок. Он заключается в том, что первый элемент помещается в середину области вывода, а место для последующих элементов освобождается путём сдвига элементов влево или вправо туда, куда выгоднее.

При каждом поступлении элементов происходит последовательно сортировка каждой половины массива с помощью бинарных вставок. Алгоритм метода бинарной вставки представляет из себя оптимизированную версию предыдущего, отличие заключается в том, что при поиске место, на которое надо вставить элемент a_i в уже упорядоченную совокупность a_0, \dots, a_{i-1} , определяется алгоритмом деления пополам (отсюда и название алгоритма "бинарные вставки" здесь понимается как "вставка делением пополам"). Метод

бинарной вставки обеспечивает более высокую скорость сортировки по сравнению с методом простой вставки.

Тестирование.

№ теста	ВВОД	ВЫВОД
	input.txt	
1	10 69 47 72 4 5 90 27 3	Your array: 10 69 47 72 4 5 90 27 3 Sorted array: 3 4 5 10 27 47 69 72 90
2	10 9 8 7 6 5 4 3 2 1	Your array: 10 9 8 7 6 5 4 3 2 1 Sorted array: 1 2 3 4 5 6 7 8 9 10
3	76 47 88 16 5 52 4 77 42 15 79 48 82 9 73 13	Your array: 76 47 88 16 5 52 4 77 42 15 79 48 82 9 73 13 Sorted array: 4 5 9 13 15 16 42 47 48 52 73 76 77 79 82 88
4	83 25 54 38 48 51 25 99 4 27 81 15 28 30 9 67 4 31 63 26 42 67 32 37 16 5 49 84 27 55 69 14 25 51 5 80 65 80 26 83 76 48 64 21 90 100 41 84 16 74	Your array: 83 25 54 38 48 51 25 99 4 27 81 15 28 30 9 67 4 31 63 26 42 67 32 37 16 5 49 84 27 55 69 14 25 51 5 80 65 80 26 83 76 48 64 21 90 100 41 84 16 74 Sorted array: 4 4 5 5 9 14 15 16 16 21 25 25 25 26 26 27 27 28 30 31 32 37 38 41 42 48 48 49 51 51 54 55 63 64 65 67 67 69 74 76 80 80 81 83 83 84 84 90 99 100
5		Error: empty array

Выводы.

При выполнении данной лабораторной работы был изучен один из алгоритмов сортировки.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>

template<typename T>
void sort_puted_value_to_the_right(T array[], int n, int gran){
    int temp;
    int right;
    int middle;
    int left;
    for (int i = gran; i < n; i++){
        if (array[i - 1] > array[i]) {
            left = gran;
            temp = array[i];
            right = i - 1 ;
            do {
                middle = (left + right) / 2;
                if (array[middle] < temp) {
                    left = middle + 1;
                }
                else {
                    right = middle - 1;
                }
            } while (left <= right);
            for (int j = i - 1; j >= left; j--){
                array[j + 1] = array[j];
            }
            array[left] = temp;
        }
    }
}

template<typename T>
void sort_puted_value_to_the_left(T array[], int n) {
    int temp;
    int right;
```

```

int middle;
int left;
for (int i = 1; i < n; i++)
    if (array[i - 1] > array[i]) {
        temp = array[i];
        left = 0;
        right = i - 1;
        do {
            middle = (left + right) / 2;
            if (array[middle] < temp) {
                left = middle + 1;
            }
            else {
                right = middle - 1;
            }
        } while (left <= right);
        for (int j = i - 1; j >= left; j--)
            array[j + 1] = array[j];
        array[left] = temp;
    }
}

template<typename T>
void sort(T array[], int n) {
    int* array_temp = new int [n];
    int count = 0;
    int temp = array[0];
    int mid_poz = 0;
    array_temp[0] = array[0];
    for (int i = 1; i < n; i++) {
        if (temp <= array[i]) {
            count++;
            array_temp[count] = array[i];
            sort_puted_value_to_the_right(array_temp, count+1,
mid_poz + 1);
        }
        if (temp > array[i]) {
            int x = array[i];

```

```

        count++;
        mid_poz++;
        for (int j = count; j > 0; j--) {
            array_temp[j] = array_temp[j - 1];
        }
        array_temp[0] = x;
        sort_puted_value_to_the_left(array_temp, mid_poz+1);
    }

}

for (int i = 0; i < n; i++) {
    array[i] = array_temp[i];
    int u = array_temp[0];
}
}

int main()
{
    int value;
    std::vector<int> tmp;
    std::ifstream file;
    std::string temp;
    file.open("input.txt");
    if (!file.is_open()) {
        std::cout << "Error: file input.txt isn't open" << std::endl;
        exit(1);
    }
    getline(file, temp);
    if (temp.empty()) {
        std::cout << "Error: empty array";
        exit(2);
    }
    while (file >> value) {
        tmp.push_back(value);
    }
    file.close();
    int count = tmp.size();
    int* array = &tmp[0];
    std::cout << "Your array: ";

```

```
    for (int i = 0; i < count; i++) {  
        std::cout << array[i] << " ";  
    }  
    sort(array, count);  
    std::cout << "\nSorted array: ";  
    for (int i = 0; i < count; i++) {  
        std::cout << array[i] << " ";  
    }  
    return 0;  
}
```

Название файла: Makefile

```
all:  
    g++ main.cpp && clear && ./a.out
```