

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ  
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №1  
по курсу «Алгоритмы и структуры данных»  
Тема: Сортировка вставками, выбором, пузырьковая  
Вариант общий

Выполнил:  
Сапожников Артём Александрович  
К3140

Проверила:  
Артамонова В.Е.

Санкт-Петербург  
2023 г.

# Содержание Отчёта

Оглавление.....	2
Основные задачи	
• Задача 1.....	3
• Задача 2.....	4
• Задача 4.....	5
• Задача 5.....	6
• Задача 6.....	7
Вывод.....	8

# Задача 1

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException; //подключаем нужные библиотеки
public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader bf = new BufferedReader(new
            FileReader(«main.txt»)); //объявляем переменную класса
        BufferedReader для чтения файла main.txt
        int n = Integer.parseInt(bf.readLine()); //считываем количество
        элементов из файла
        String s = bf.readLine() + « «; //обрабатываем строку элементов и
        добавляем дополнительный пробел
        long []arr = new long[n]; //создаём массив на n элементов
        for (int i = 0; i < n; i++) { //заполняем массив
            arr[i] = Integer.parseInt(s.substring(0,s.indexOf(« «))); //в
            массив записываем подстроку начиная от начала строки до первого пробела
            s = s.substring(s.indexOf(« «)+1); //от основной строки отрезаем
            начало до первого пробела
        }
        for (int i = 1; i < n; i++) { //последовательно рассматриваем каждый
        элемент начиная со второго
            for (int j = i-1; j >= 0; j--) { //последовательно рассматриваем
            каждый элемент вплоть до i, начиная с i-1
                if(arr[j+1]<arr[j]) { //если 2 элемента не находятся в порядке
                возрастания
                    long buffer = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = buffer; //меняем значение arr[j] и arr[j+1]
                }
            }
        }
        for (int i = 0; i < n; i++) { //выводим элементы отсортированного
        массива
            System.out.print(arr[i] + « «);
        }
        System.out.println();
    }
}
```

input	6	output	26 31 41 41 58 59
	31 41 59 26 41 58		

	Время выполнения (мс)	Затраты памяти(кб)
Значения нижней границы	5,84	3695
Пример 1	6,22	3695
Верхняя граница	24,37	9014

## Задача 2

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException; //подключаем нужные библиотеки
public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader bf = new BufferedReader(new
            FileReader("main.txt")); //объявляем переменную класса BufferedReader
//для чтения файла main.txt
        int n = Integer.parseInt(bf.readLine()); //считываем количество элементов из
//файла
        String s = bf.readLine() + " "; //обрабатываем строку элементов и добавляем
//дополнительный пробел
        long []arr = new long[n]; //создаём массив на n элементов
        for (int i = 0; i < n; i++) { //заполняем массив
            arr[i] = Integer.parseInt(s.substring(0, s.indexOf(" "))); //в массив
//записываем подстроку начиная от начала строки до первого пробела
            s = s.substring(s.indexOf(" ") + 1); //от основной строки отрезаем начало до
//первого пробела
        }
        int []index = new int[n]; //создаём массив индексов
        index[0] = 1; //очевидно, первый индекс равен 1
        for (int i = 1; i < n; i++) { //последовательно рассматриваем каждый элемент
//начиная со второго
            for (int j = i - 1; j >= 0; j--) { //последовательно рассматриваем каждый
//элемент вплоть до i, начиная с i-1
                if (arr[j + 1] < arr[j]) { //если 2 элемента не находятся в порядке
//возрастания
                    long buffer = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = buffer; //меняем значение arr[j] и arr[j + 1] через буффер
                    if (j == 0) //если j дошло до 0, значит сортировка дошла до начала
//массива
                        index[i] = 1; //поэтому индекс будет равен 1
                } else {
                    index[i] = j + 2; //j - позиция ближайшего к началу элемента пары,
//поэтому в массив индексов запишем j + 2, а не j + 1
                    break; //в противном случае прерываем цикл (элементы левее
//рассортированы)
                }
            }
        }
        for (int i = 0; i < n; i++) { //выводим элементы массива индексов
            System.out.print(index[i] + " ");
        }
        System.out.println();
        for (int i = 0; i < n; i++) { //выводим элементы отсортированного массива
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
}
```

input	10	output	1 2 2 2 3 5 5 6 9 1
	1 8 4 2 3 7 5 6 9 0		0 1 2 3 4 5 6 7 8 9

	Время выполнения (мс)	Затраты памяти(кб)
Значения нижней границы	9,47	4096
Пример 1	10,24	4096
Верхняя граница	34,36	9014

## Задача 4

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException; //подключаем нужные библиотеки
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader bf = new BufferedReader(new
            FileReader("main.txt")); //объявляем переменную класса
        //BufferedReader для чтения файла main.txt
        int n = Integer.parseInt(bf.readLine()); //считываем количество
        //элементов из файла
        String s = bf.readLine() + " "; //обрабатываем строку элементов и
        //добавляем дополнительный пробел
        long []arr = new long[n]; //создаём массив на n элементов
        for (int i = 0; i < n; i++) { //заполняем массив
            arr[i] = Integer.parseInt(s.substring(0, s.indexOf(" "))); //в
            //массив записываем подстроку начиная от начала строки до первого пробела
            s = s.substring(s.indexOf(" ")+1); //от основной строки отрезаем
            //начало до первого пробела
        }
        int v = Integer.parseInt(bf.readLine()); //считываем искомое число из
        //файла
        ArrayList<Integer> arrayList = new ArrayList<>(); //объявляем список
        //индексов найденных элементов
        int flag = 0; //объявляем переменную flag
        for (int i = 0; i < n; i++) { //пробегаем по массиву
            if(arr[i]==v) { //если элемент равен искомому числу
                arrayList.add(i); //записываем их в лист
                flag = 1; //если искомое число найдено, то flag поднят
            }
        }
        if(flag==0) //если flag опущен
            System.out.println(-1); //выводим -1
        else { //если flag поднят
            System.out.println(arrayList.size()); //размер листа = количеству
            //найденных значений
            for (int i = 0; i < arrayList.size(); i++) { //выводим все
            //значения из списка
                System.out.print(arrayList.get(i) + " ");
            }
            System.out.println();
        }
    }
}
```

input	5	output	2
	1 2 3 2 1		1 3
	2		

	Время выполнения (мс)	Затраты памяти(кб)
Значения нижней границы	1,58	3695
Пример 1	1,78	3695
Верхняя граница	6,74	8604

## Задача 5

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException; //подключаем нужные библиотеки

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader bf = new BufferedReader(new
            FileReader("main.txt")); //объявляем переменную класса
        //BufferedReader для чтения файла main.txt
        int n = Integer.parseInt(bf.readLine()); //считываем количество
        //элементов из файла
        String s = bf.readLine() + " "; //обрабатываем строку элементов и
        //добавляем дополнительный пробел
        long []arr = new long[n]; //создаём массив на n элементов
        for (int i = 0; i < n; i++) { //заполняем массив
            arr[i] = Integer.parseInt(s.substring(0, s.indexOf(" "))); //в
            //массив записываем подстроку начиная от начала строки до первого пробела
            s = s.substring(s.indexOf(" ")+1); //от основной строки отрезаем
            //начало до первого пробела
        }
        for (int i = 0; i < n; i++) {
            long min = 1000000001; //создаём переменную min
            int index = 0; //и переменную index
            for (int j = i; j < n; j++) { //находим min элемент в области от i
            //до n
                if (min > arr[j]) {
                    min = arr[j];
                    index = j; //записываем его индекс в переменную index
                }
            }
            long buffer = arr[i]; //меняем местами элементы arr[i] и
            //arr[index]
            arr[i] = min;
            arr[index] = buffer;
        }
        for (int i = 0; i < n; i++) { //выводим массив
            System.out.print(arr[i] + " ");
        }
    }
}
```

input	5 1 5 4 6 2	output	1 2 4 5 6
-------	----------------	--------	-----------

	Время выполнения (мс)	Затраты памяти(кб)
Значения нижней границы	5,9	3695
Среднее значение	24,67	9014
Верхняя граница	27,8	9014

## Задача 6

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException; //подключаем нужные библиотеки

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader bf = new BufferedReader(new
            FileReader("main.txt")); //объявляем переменную класса
        //BufferedReader для чтения файла main.txt
        int n = Integer.parseInt(bf.readLine()); //считываем количество
        //элементов из файла
        String s = bf.readLine() + " "; //обрабатываем строку элементов и
        //добавляем дополнительный пробел
        long []arr = new long[n]; //создаём массив на n элементов
        for (int i = 0; i < n; i++) { //заполняем массив
            arr[i] = Integer.parseInt(s.substring(0, s.indexOf(" "))); //в
            //массив записываем подстроку начиная от начала строки до первого пробела
            s = s.substring(s.indexOf(" ")+1); //от основной строки отрезаем
            //начало до первого пробела
        }
        int flag = 0; //объявляем переменную flag
        while (flag==0) { //если флаг опущен сортировка продолжается
            flag = 1; //флаг поднимаем
            for (int i = 0; i < n-1; i++) {
                if(arr[i]>arr[i+1]) { //если в массиве есть неотсортированные
                //элементы
                    long buffer = arr[i]; //меняем их местами
                    arr[i] = arr[i + 1];
                    arr[i + 1] = buffer;
                    flag = 0; //сортировка ДОЛЖНА продолжаться
                }
            }
        }
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

input	5 1 5 4 6 2	output	1 2 4 5 6
-------	----------------	--------	-----------

	Время выполнения (мс)	Затраты памяти(кб)
Значения нижней границы	5,23	3695
Среднее значение	25,28	9014
Верхняя граница	28,33	9014

## Вывод

В данной лабораторной работе были рассмотрены простейшие методы сортировки за  $n^2$ . Эти алгоритмы применимы для массивов с небольшим количеством элементов (до 1000) и совершенно не применимы для огромных массивов. Если проводить аналогию с ЕГЭ по информатике, то для задачи 27а такие методы сортировки были уместны, а для пункта б время выполнения программы превышало 4 часа. Практика показала, что сортировка вставками была немного быстрее сортировки выбором и сортировки пузырьком (24,37 мс против 27,8 и 28,33 соответственно). Связано это с тем, что сортировка вставками алгоритмически немного проще своих конкурентов, для её выполнения не требуются дополнительные переменные. Еще в лабораторной был линейный поиск: опять же, простейший алгоритм применимый для небольших массивов. Однако для более серьезных задач алгоритмы с временем  $n^2$  не пригодны, для них нужны более хитрые алгоритмы.