

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
Кафедра вычислительной техники

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 3**

Поиск.  
Вариант № 10

Преподаватель

\_\_\_\_\_  
подпись, дата

Матковский И.В.

Студент КИ18-096, 031830645

\_\_\_\_\_  
подпись, дата

Котов С.А.

Красноярск 2019

## 1 Задание

Разработать для решения поставленной задачи алгоритм; решить поставленную задачу с помощью последовательного поиска и поиска, предусмотренного вариантом. Оценить алгоритмы.

Первый метод поиска: «Последовательный поиск», второй метод поиска: «Двоичный поиск».

## 2 Код программы

```
1 int searchIncremental(vector<int>& sequence, int key) {
2     int min = abs(sequence[0] - key);
3     int ch = NULL;
4     for (int i = 0; i < sequence.size(); i++) {
5         if (abs(sequence[i] - key) <= min) {
6             min = abs(sequence[i] - key);
7             ch = sequence[i];
8         }
9     }
10    }
11    return ch;
12 }
13
14 int searchBinary(vector<int> &sequence, int key) {
15     int midd = 0;
16     int left = 0;
17     int right = sequence.size() - 1;
18
19     while (true) {
20         for (int i = 0; i < right; i++) {
21             int midd = (right + left) / 2;
22             if (sequence[midd] > key) {
23                 right = midd;
24             }
25             else {
26                 left = midd;
27             }
28         }
29     }
30     if (left != right) {
31         return (abs(sequence[right] - key) < abs(sequence[left] - key) ? right : left);
32     }
33     return left;
34 }
35 }
```

### 3 Теоретические оценки сложности алгоритмов

Временная сложность «Последовательного поиска» равна  $O(\text{sequence.size}())$ , а у «Двоичного поиска» временная сложность равна  $O(\log(\text{sequence.size}()))$ .

«Последовательный поиск» перебирает все элементы в векторе, сравнивая их с заданным ключом.

В «Двоичном поиске» искомый ключ сравнивается с ключом среднего элемента в массиве. Если они равны, то поиск успешен. В противном случае поиск осуществляется аналогично в левой или правой частях массива.

### 4 Экспериментальные оценки сложности алгоритмов

sequence.size()	Время 1, сек	Время 2, сек
100	0	0
50000	0	0
1000000	1	0
10000000	5	0