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 \_\_\_\_\_\_\_\_ Матковский И.В.

подпись, дата

Студент КИ18-09б, 031830645 \_\_\_\_\_\_\_\_ Котов С.А.

подпись, дата

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 поиск: «Двоичный поиск».

**2 Код программы**

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

**3 Теоретические оценки сложности алгоритмов**

Временная сложность «Последовательного поиска» равна O(sequence.size()), а у «Двоичного поиска» временная сложность равна O(log(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 |