

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФГБОУ ВО «Брянский государственный технический университет»

Факультет информационных технологий.

Кафедра «Информатика и программное обеспечение»

Отчет по лабораторной работе

«Алгоритмы быстрого поиска»

Выполнил:

студент гр. О-18-ПРИ-РПС-Б

Подгорняк А.А.

Проверил:

к.т.н., проф. В. К. Гулаков

Брянск 2020

**СОДЕРЖАНИЕ**

[1. ЗАДАНИЕ 3](#_Toc43846773)

[2. Описание программы 4](#_Toc43846774)

[3. Демонстрация программы 6](#_Toc43846775)

## ЗАДАНИЕ

Имеется n монет, среди которых одна фальшивая (легче или тяжелее других монет). Дополнительно дается одна настоящая. Имеются весы, с помощью которых можно сравнить различные группы монет. Разработать алгоритм и программу, позволяющую при минимальном числе сравнений определить фальшивую монету (если она есть).

## Описание программы

Программа представляет собой консольное приложение, которое решает поставленную задачу

Все монеты хранятся в векторе. Для поиска используется идея бинарного поиск,но сравниваются не отдельные элементы, а сумма элементов 2 половин вектора.

|  |
| --- |
| int search(vector <coin >& a)  {  int result=0;  int sum1=0, sum2=0;  if (a.size() % 2 == 0)  {  if ( a.size() > 2)  {  for (int i = 0; i < a.size() / 2; i++)  {  sum1 += a[i].weight;  }  for (int i = a.size() / 2; i < a.size() ; i++)  {  sum2 += a[i].weight;  }  }  else  {  result = compare(a[0].weight, a[1].weight);  if (result == 1)  return a[1].pos;  else if (result == 2)  return a[0].pos;  else  return -1;  }  result = compare(sum1, sum2);  if (result == 1)  {  auto begin = a.begin();  auto end = a.end();  a.erase(begin, end- a.size()/2);  return search(a);  }  else if (result == 2)  {  auto begin = a.begin();  auto end = a.end();  a.erase(begin+ a.size() / 2, end );  return search(a);  }  else if (result == 3)  return -1;  }  else if (a.size() % 2 != 0)  {  if (a.size() == 1)  {  cout << "Невозможно определить" << endl;  return -2;  }  else  {  result = compare(a[a.size() - 2].weight, a[a.size() - 1].weight);  if (result == 3)  a.pop\_back();  else if (result == 1)  return a[a.size() - 1].pos;  else if (result == 2)  return a[a.size() - 2].pos;  return search(a);  }  }  } |

Листинг

При запуске программы пользователю предлагается ввести количество монет:

Фальшивая монета выбирается случайным образом, и ей присваивается вес меньший, чем у оригинала. После выполнения алгоритма результат сравнивается с изначально записанным расположением фальшивой монеты.

|  |
| --- |
| * cout << "Введите количество монет " ; * cin >> n; * int fake; * fake = rand()%n; * vector <coin> a(n); * int weight = 2, Fweight = 1; * for (int i = 0; i < n;i++) * { * if (i == fake) * { * a[i].weight = Fweight; * a[i].pos = i; * } * else * { * a[i].weight = weight; * a[i].pos = i; * } * } * found = search(a); * if (found == -1) * cout << "Фальшивой монеты нет" << endl; * else if (found>-1) * cout << "Фальшивая монета лежит на позиции " << fake << endl << "Результат алгоритма " << found << endl; |

Листинг

## Демонстрация программы



Рис. . Пример работы программы

Как мы можем видеть, все работает корректно