МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

**ОТЧЁТ**

**«ЛАБОРАТОРНАЯ №12:**

**ЛИНЕЙНЫЙ И ИНТЕРПОЛЯЦИОННЫЙ ПОИСКИ»**

Дисциплина: «Программирование»

Выполнил:

Студент группы ИВТ-21-2б

Безух Владимир Сергеевич

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь, 2022

Содержание

[1. Постановка задачи 3](#_Toc98243474)

[2. Анализ задачи 4](#_Toc98243475)

[3. Исходный код 5](#_Toc98243476)

[4. Анализ результатов 7](#_Toc98243477)

# Постановка задачи

Найти индекс значения по заданному ключу с помощью линейного и интерполяционного поисков.

# 

# Анализ задачи

Линейный поиск: поиск значения осуществляется простым сравнением очередного рассматриваемого значения и, если значения совпадают (с той или иной точностью), то поиск считается завершённым. Асимптотическая сложность алгоритма O(n).

Интерполяционный поиск. Алгоритм производит предсказание местонахождения элемента: поиск происходит подобно двоичному поиску, но вместо деления области поиска на две части, интерполяционный поиск производит оценку новой области поиска по расстоянию между ключом и текущим значением элемента. В среднем случае асимптотическая сложность равна log(log(n)). В плохом случае (например, когда значения элементов экспоненциально возрастают) асимптотическая сложность вырождается в O(n).

# 

# Исходный код

#include <vector>

#include <iostream>

void printVector(const std::vector<int>& arr)

{

for (const auto& value : arr)

std::cout << value << ' ';

std::cout << '\n';

}

void printArray(const int\* array, const size\_t& size)

{

for (size\_t i = 0; i != size; ++i)

std::cout << i << " <- индекс | значение -> " << array[i] << '\n';

}

void linearSearch(const std::vector<int>& arr, std::vector<int>& result, const int& key)

{

for (size\_t i = 0; i != arr.size(); ++i)

if (arr[i] == key)

result.emplace\_back(i);

}

int interpolationSearch(const int\* array, const size\_t& size, int key)

{

int left = 0, right = static\_cast<int>(size) - 1, middle;

while (array[left] < key && key <= array[right])

{

middle = left + ((key - array[left])

\* (right - left)) / (array[right] - array[left]);

if (array[middle] < key)

left = middle + 1;

else if (array[middle] > key)

right = middle - 1;

else

return middle;

}

if (array[left] == key)

return left;

return -1; // value is not found

}

void firstTask()

{

std::vector<int> arr, result;

for (int i = 0; i != 10; ++i)

arr.emplace\_back(i \* i);

for (int i = 0; i != 10; ++i)

arr.emplace\_back(i \* i);

printVector(arr);

int key; std::cin >> key;

linearSearch(arr, result, key);

printVector(result);

if (result.size())

for (const auto& index : result)

std::cout << "Ключ " << key << " находится по индексу "

<< index << ": " << arr[index] << '\n';

else

std::cout << "Ключ " << key << " не найден в массиве" << '\n';

}

void secondTask()

{

const size\_t size = 10;

int array[size] = {0, 1, 4, 9, 16, 25, 36, 49, 64, 81};

int key, index;

std::cin >> key;

printArray(array, size);

index = interpolationSearch(array, size, key);

if (index != -1)

std::cout << "Значение " << key << " найдено по индексу " << index;

else

std::cout << "Значение не найдено :(";

}

int main()

{

setlocale(LC\_ALL, "Russian");

firstTask();

secondTask();

}

# Анализ результатов

Результаты работы программы (рис. 1).

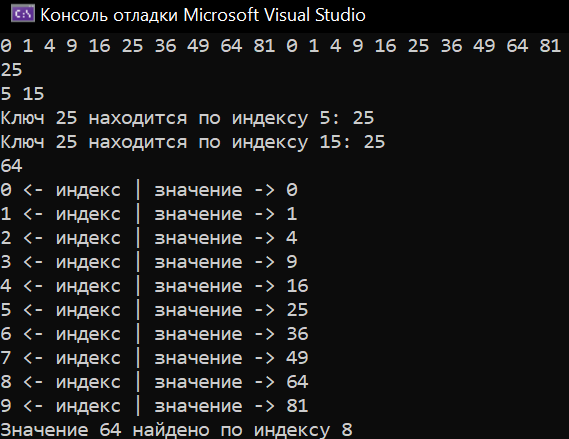


Рисунок 1 — Результаты