Министерство высшего образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ

Дисциплина: Информатика

Тема: Линейный поиск

Выполнил

Студент группы РИС-22-2б

Худеньких В.Д.

Проверил

доц. кафедры ИТАС

Полякова О.А.

Пермь 2023

**Содержание**

Введение **3**

Разработка программы вычисления значений функции с использованием языка программирования C++**9**

Заключение**11**

**Введение**

Одним из базовых алгоритмов поиска в информатике является линейный поиск. Он используется для поиска ключевого элемента в неупорядоченном массиве данных. Алгоритм реализуется последовательным проходом по массиву и сравнением каждого элемента с искомым ключом до тех пор, пока не будет найден элемент с этим ключом или не будет достигнут конец массива.

В данной лабораторной работе была поставлена задача реализации линейного поиска в языке программирования С++. Цель работы - изучить основные принципы работы алгоритма и разработать программу, реализующую линейный поиск в заданном наборе данных.

Эффективность линейного поиска определяется количеством сравнений элементов с ключом, которые приходится выполнять при поиске. В случае, если в массиве находится много элементов, линейный поиск может потребовать значительного количества времени на поиск. Поэтому, возникает необходимость использования более эффективных алгоритмов поиска, таких как бинарный поиск или интерполяционный поиск.

**Разработка программы с использованием языка программирования C++.**

Для решения задачи была разработана программа, основанная на алгоритме линейного поиска. Программа производит поиск элемента в массиве, представляющим собой простой список, и выводит на экран индекс элемента в массиве. Если элемент не найден, программа выводит соответствующее сообщение.

Для этого была создана функция "linear\_search", которая принимает на вход указатель на массив данных и размер массива, а также значение ключа, который нужно найти. Функция проходит по всем элементам массива и сверяет их значение с заданным ключом. Если элемент, равный ключу, найден, функция возвращает его индекс, а если нет - возвращает значение "-1".

В программе также были использованы функции для генерации и вывода массива на экран, что позволяет пользователю убедиться в корректности поиска.

Были проведены тесты программы на различных массивах разного размера, содержащих различные значения. В ходе тестирования было установлено, что программа работает корректно и успешно находит заданный элемент в массиве.

Однако, было также выявлено, что линейный поиск не является оптимальным алгоритмом для работы с большими объёмами данных. В случае, если количество элементов в массиве очень большое, время работы линейного поиска может значительно увеличиться, что приведет к плохой производительности программы. Поэтому, если требуется работать с большими объёмами данных, целесообразно использовать более эффективные алгоритмы поиска.

**Заключение**

В результате выполнения лабораторной работы была разработана программа, реализующая линейный поиск в массиве данных. Был изучен алгоритм линейного поиска и разработана соответствующая функция на языке С++. Проведено тестирование программы на различных входных данных.

Полученные результаты показали, что линейный поиск является простым и эффективным алгоритмом для решения простых задач поиска в небольших массивах. Однако, в случае, если количество элементов в массиве существенно больше, линейный поиск может стать нерентабельным по времени выполнения. В таких случаях более эффективно использовать другие алгоритмы поиска, например, бинарный поиск или интерполяционный поиск.

Таким образом, линейный поиск - это базовый алгоритм в информатике, который должен быть изучен для освоения основ программирования. В то же время, при решении реальных задач, необходимо выбирать оптимальный алгоритм в зависимости от конкретной ситуации и объёма данных.

**ПРИЛОЖЕНИЕ А**

**Блок-схемы программ**

