

БУ ВО «Сургутский государственный университет»
Политехнический институт
Кафедра автоматизированных систем обработки информации и управления

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
ПО ТЕМЕ «Базовые алгоритмы поиска (линейный и двоичный)»
ПО ДИСЦИПЛИНЕ «Поисковые алгоритмы в информационном
пространстве»

Выполнил: студент группы №606-12,
Речук Дмитрий Максимович
Принял: ст. преподаватель,
Гавриленко Анна Владимировна

Сургут 2024

Данная лабораторная работа посвящена изучению и сравнению двух алгоритмов поиска: линейного поиска и двоичного поиска. Основная цель работы — исследовать их временную сложность, провести эксперименты на массивах различных размеров и на основе полученных данных построить графики для наглядного сравнения.

Поиск является одной из базовых операций в алгоритмах и структурах данных, широко используемой в различных областях, таких как базы данных, компьютерное зрение, машинное обучение и другие. Выбор метода поиска зависит от структуры данных, их размера и необходимости оптимизации по времени выполнения.

Алгоритмы

1. Линейный поиск

Линейный поиск — это простой метод поиска элемента в массиве, при котором программа последовательно перебирает элементы массива, сравнивая их с искомым значением.

Особенности:

- Не требует предварительной сортировки массива.
- В худшем случае требуется проверить каждый элемент массива.
- Подходит для небольших массивов или не отсортированных данных.
- Временная сложность: $O(n)$, где n — количество элементов в массиве.

Пример работы линейного поиска:

Допустим, у нас есть массив $[3, 7, 1, 9, 5]$, и мы ищем число 9. Линейный поиск пройдет по каждому элементу: $3 \rightarrow 7 \rightarrow 1 \rightarrow 9$ и найдет значение на четвертой позиции.

2. Двоичный поиск

Двоичный поиск является более эффективным методом поиска, но требует предварительной сортировки массива. Он работает по принципу деления массива пополам, сокращая количество проверяемых элементов на каждом шаге.

Особенности:

- Работает только на отсортированных массивах.

- На каждом шаге сравнивает искомый элемент с центральным элементом массива.
- Если элемент меньше центрального, поиск продолжается в левой половине массива, иначе — в правой.
- Временная сложность: $O(\log n)$, что делает его значительно быстрее линейного поиска при больших объемах данных.

Пример работы двоичного поиска:

Допустим, у нас есть отсортированный массив $[1, 3, 5, 7, 9]$, и мы ищем число 7. Двоичный поиск начинает с центрального элемента (5), сравнивает его с 7 и переходит к правой части (7, 9). Затем сравнивает 7 с центральным элементом (7) и находит его.

Требования

Для выполнения работы потребуется установленный Python версии 3.x и библиотека matplotlib, которая будет использована для построения графиков.

Установка зависимостей

```
pip install matplotlib
```

Использование

1. Запустите скрипт, выполнив следующую команду:

```
python search_comparison.py
```

2. В консоли отобразятся результаты времени выполнения для каждого алгоритма на массивах разного размера.
3. График зависимости времени выполнения от размера массива будет построен и отображен в отдельном окне.

Пример вывода

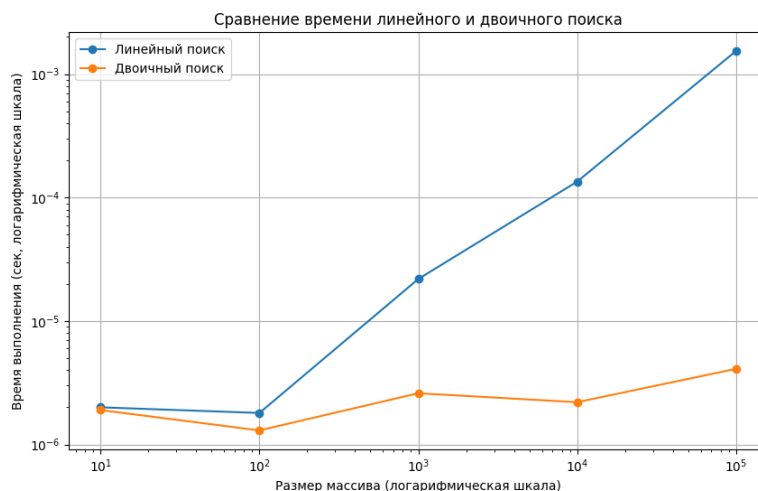
Размер массива	Линейный поиск (сек)	Двоичный поиск (сек)
10	0.00000080	0.00000040
100	0.00000220	0.00000060
1000	0.00001850	0.00000070
10000	0.00017230	0.00000110
100000	0.00179500	0.00000130

Анализ результатов

На основании проведенных тестов можно сделать следующие наблюдения: - Линейный поиск демонстрирует линейный рост времени выполнения ($O(n)$). Это означает, что при увеличении размера массива время поиска возрастает пропорционально количеству элементов. - Двоичный поиск имеет логарифмическую сложность ($O(\log n)$). Это делает его гораздо более эффективным на больших массивах, так как рост времени выполнения значительно медленнее по сравнению с линейным поиском. - На массиве из 100 000 элементов разница в скорости выполнения становится особенно заметной: двоичный поиск выполняется в сотни раз быстрее.

Графический анализ

График, построенный по экспериментальным данным, подтверждает теоретические оценки сложности алгоритмов: - График времени выполнения линейного поиска представляет собой почти прямую линию, указывающую на линейный рост. - График двоичного поиска имеет более пологий характер, что соответствует логарифмической зависимости.



Выводы

- Двоичный поиск является значительно более эффективным при работе с большими отсортированными массивами.
- Линейный поиск остается полезным для небольших массивов или в случаях, когда данные не отсортированы, а их предварительная сортировка займет слишком много времени.

- Выбор метода поиска должен основываться на размерах данных и требованиях к скорости выполнения.

Таким образом, использование двоичного поиска предпочтительно для работы с большими объемами данных, если они уже отсортированы или могут быть предварительно упорядочены. В противном случае линейный поиск остается универсальным, хоть и менее эффективным методом поиска.