



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**Лабораторная работа № 3
по дисциплине «Анализ алгоритмов»**

Тема Поиск в массиве

Студент Равашдех Ф.Х.

Группа ИУ7-55Б

Преподаватели Строганов Ю.В.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитическая часть	4
1.1 Поиск полным перебором	4
1.2 Двоичный поиск	4
2 Конструкторская часть	5
2.1 Поиск полным перебором	5
2.2 Двоичный поиск	6
3 Технологическая часть	7
3.1 Требования к программному обеспечению	7
3.2 Средства реализации	7
3.3 Реализация алгоритмов	7
3.4 Тестирование	8
4 Исследовательская часть	9
4.1 Технические характеристики	9
4.2 Количество сравнений	9
4.3 Вывод	11
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	13

ВВЕДЕНИЕ

Задача поиска в массиве элемента с заданным значением встречается в программировании очень часто.

Цель лабораторной работы — сравнение алгоритмов нахождения значения в массиве методом полного перебора и методом двоичного поиска. Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) реализовать алгоритмы указанных методов;
- 2) провести замеры количества сравнений в реализованных алгоритмах от искомого значения;
- 3) провести сравнительный анализ реализаций алгоритмов по полученным экспериментальным данным;
- 4) описать результаты в отчете.

1 Аналитическая часть

1.1 Поиск полным перебором

При использовании метода поиска полным перебором происходит последовательный просмотр элементов массива и сравнении каждого из них с образцом для поиска [1].

Для алгоритма нахождения искомого значения в массиве полным перебором количество исходов равно $N + 1$: искомое значение не найдено и N возможных случаев расположения элемента в массиве. Лучший случай: за одно сравнение ключ найден в начале массива. Худших случаев два: за N сравнений либо элемент не найден, либо элемент найден на последнем сравнении.

1.2 Двоичный поиск

При использовании метода двоичного поиска предполагается, что массив в котором осуществляется поиск уже упорядочен. Вводятся левая и правая граница поиска. Выбирается центральный элемент в области между границами поиска и сравнивается с искомым значением. Если искомое значение меньше центрального элемента, то центральный элемент становится правой границей, если искомое значение больше центрального элемента, то центральный элемент становится левой границей, в оставшемся случае искомое значение найдено. Сравнения повторяются пока границы не сойдутся или искомое значение будет найдено. Если левая граница станет больше правой, а искомое значение не будет найдено, то его нет в массиве [2].

Лучший случай: за одно сравнение ключ найден в середине массива. Худших случаев два: за $\lceil \log_2 N \rceil$ сравнений либо элемент не найден, либо элемент найден на последнем сравнении.

Вывод

В данном разделе рассмотрены алгоритмы нахождения искомого значения в массиве.

2 Конструкторская часть

2.1 Поиск полным перебором

Схема алгоритма нахождения искомого значения полным перебором представлена на рисунке 2.1.

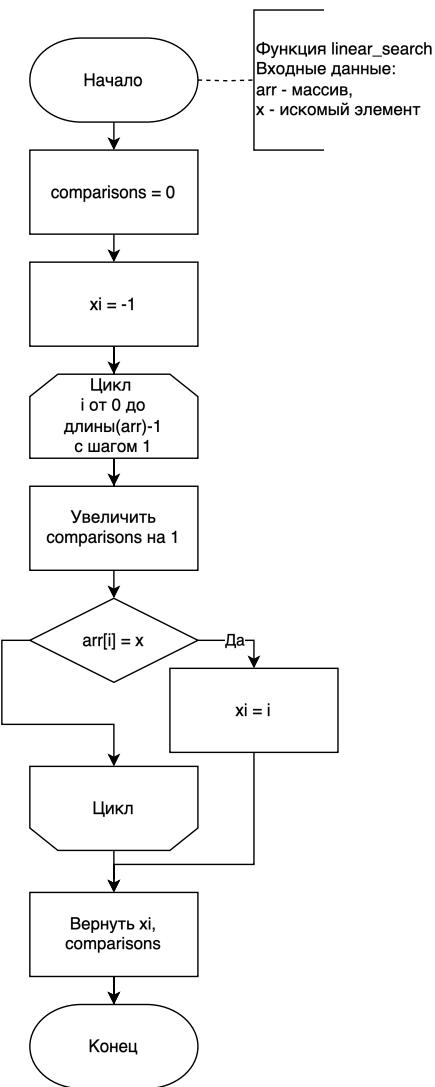


Рисунок 2.1 — Схема алгоритма нахождения искомого значения полным перебором

2.2 Двоичный поиск

Схема алгоритма нахождения искомого значения двоичным поиском представлена на рисунке 2.2.

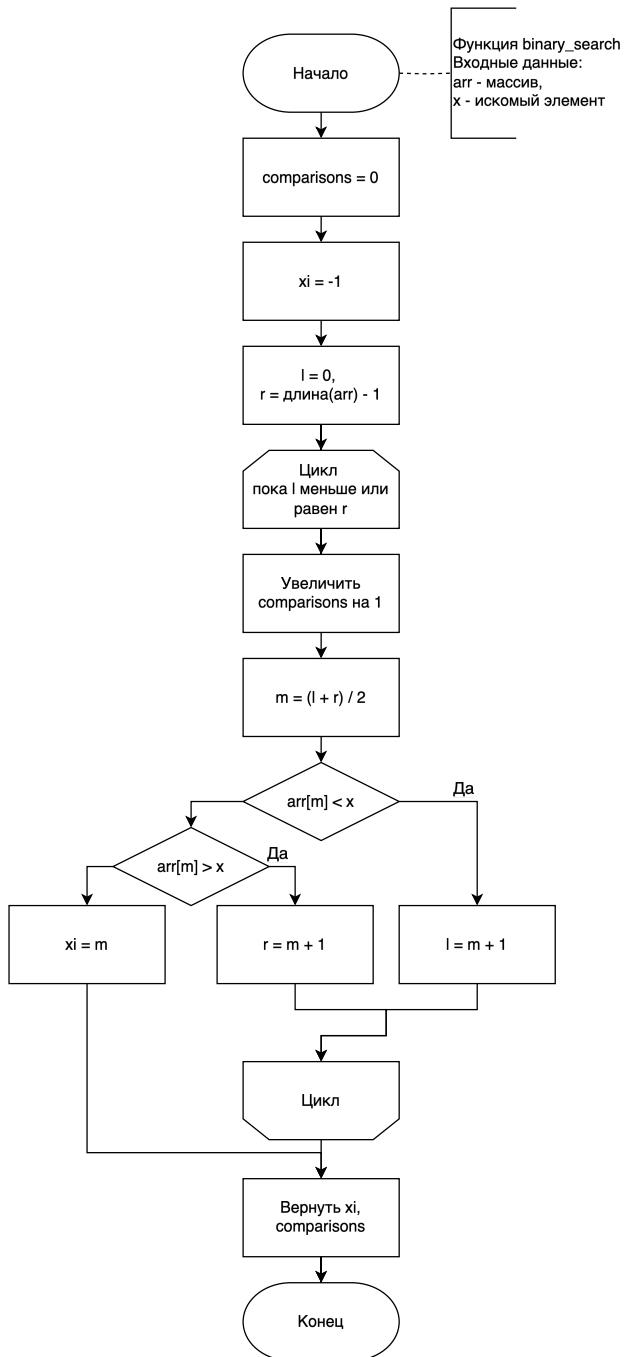


Рисунок 2.2 — Схема алгоритма нахождения искомого значения двоичным поиском

Вывод

В данной части работы были описаны алгоритмы нахождения искомого значения полным перебором и двоичным поиском.

3 Технологическая часть

3.1 Требования к программному обеспечению

Входные данные: массив и искомое значение;

Выходные данные: позиция искомого значения в массиве и количество сравнений.

3.2 Средства реализации

Алгоритмы для данной лабораторной работы были реализованы на языке *Python* [3], который имеет библиотеку для визуализации данных *Matplotlib* [4].

3.3 Реализация алгоритмов

Реализация алгоритмов нахождения искомого значения полным перебором и двоичным поиском представлена в листингах (3.1) и (3.2).

Листинг 3.1 — Функция нахождения искомого значения полным перебором

```
def linear_search(arr, x):
    comparisons, xi = 0, -1
    for i in range(len(arr)):
        comparisons += 1
        if arr[i] == x:
            xi = i
            break
    return xi, comparisons
```

Листинг 3.2 — Функция нахождения искомого значения двоичным поиском

```
def binary_search(arr, x):
    comparisons, xi = 0, -1
    l, r = 0, len(arr) - 1
    while l <= r:
        m = (l + r) // 2
        comparisons += 1
        if arr[m] < x:
            l = m + 1
        elif arr[m] > x:
            r = m - 1
        else:
            xi = m
            break
    return xi, comparisons
```

3.4 Тестирование

В таблице 3.1 представлены данные о результатах тестирования реализации алгоритмов нахождения искомого значения полным перебором и двоичным поиском.

Таблица 3.1 — Модульные тесты

Входные данные		Алгоритм, индекс и количество сравнений		
Массив	Искомое значение	Полный перебор		Двоичный поиск
arr {...}	34772	454	455	331 9
arr {...}	34773	-1	1033	-1 10
arr {...}	999	-1	1033	-1 10

Все тесты пройдены успешно.

Вывод

В данном разделе были рассмотрены требования к программному обеспечению, используемые средства реализации, приведена реализация алгоритмов и результаты тестирования.

4 Исследовательская часть

4.1 Технические характеристики

Характеристики устройства, на котором выполнялись замеры:

- 1) операционная система — macOS Sonoma 14.1 (23B2073);
- 2) процессор — Apple M3;
- 3) оперативная память — 16 Гб.

4.2 Количество сравнений

Согласно варианту задания, массив состоит из $N = 1033$ элементов. Оценка трудоемкости алгоритма измерена в количестве сравнений, потребовавшихся для поиска искомого значения в массиве в каждом из $N + 1$ случаев (N искомых значений, существующих в массиве, и одно искомое значение, которого нет в массиве).

На рисунке 4.1 показана гистограмма зависимости количества сравнений от индекса искомого значения в массиве при использовании алгоритма нахождения искомого значения полным перебором.

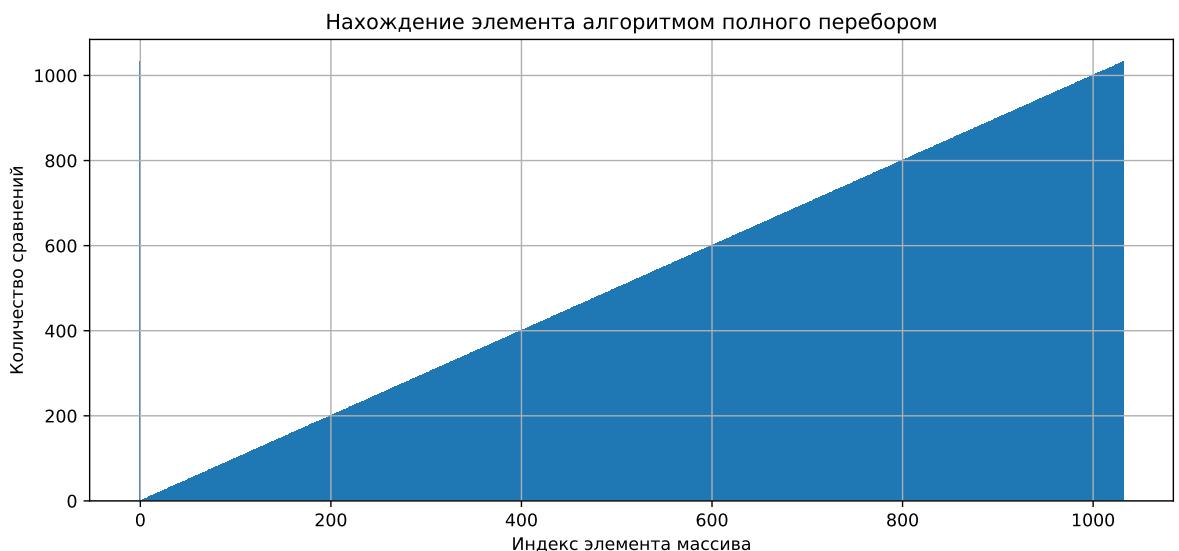


Рисунок 4.1 — Гистограмма для алгоритма полным перебором

На рисунке 4.2 показана гистограмма зависимости количества сравнений от индекса искомого значения в массиве при использовании алгоритма нахождения искомого значения двоичным поиском.

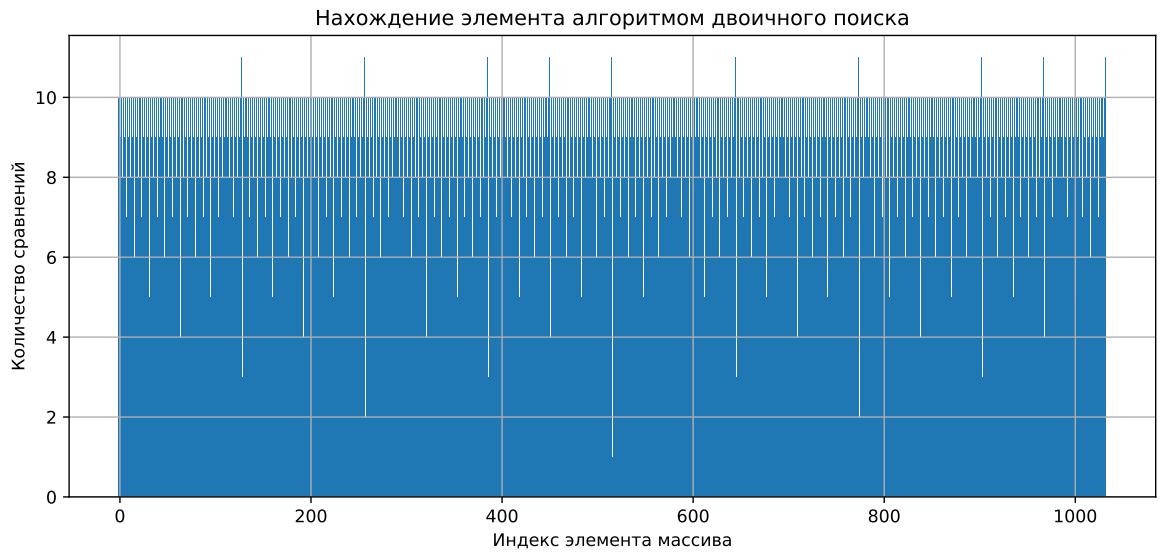


Рисунок 4.2 — Гистограмма для алгоритма двоичным поиском

На рисунке 4.3 показана гистограмма количества сравнений при использовании алгоритма нахождения искомого значения двоичным поиском, отсортированная по возрастанию количества сравнений

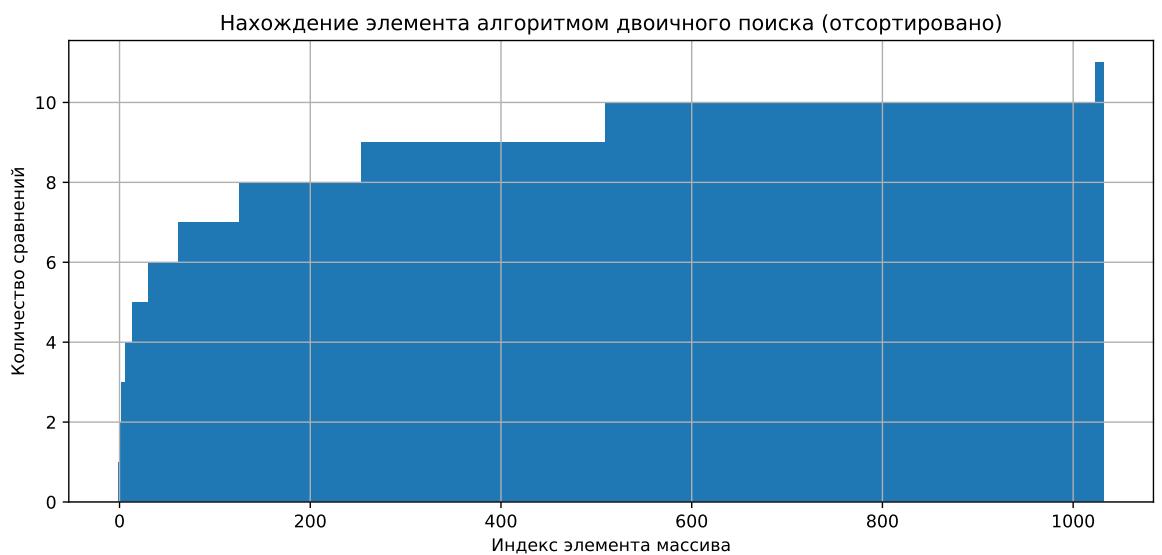


Рисунок 4.3 — Гистограмма для алгоритма двоичным поиском отсортированная по возрастанию количества сравнений

4.3 Вывод

В данном разделе были приведены технические характеристики и время выполнения алгоритмов. Количество сравнений в алгоритме полным перебором зависит от индекса искомого элемента и может достигать N в худших случаях. В алгоритме двоичного поиска количество сравнений в худших случаях равно $\lceil \log_2 N \rceil$, что для $N = 1033$ равно 11. Это означает, что алгоритм двоичного поиска в большинстве случаев оказывался быстрее алгоритма полным перебором. Недостатком алгоритма двоичного поиска является необходимость в сортировке массива перед поиском.

ЗАКЛЮЧЕНИЕ

Цель работы достигнута: сравнение алгоритмов нахождения искомого элемента в массиве полным перебором и двоичным поиском было проведено.

В ходе выполнения лабораторной работы были решены все задачи:

- 1) реализованы алгоритмы указанных методов;
- 2) проведены замеры количества сравнений в реализованных алгоритмах от искомого значения;
- 3) проведен сравнительный анализ реализаций алгоритмов по полученным экспериментальным данным;
- 4) описаны результаты в отчете.

Сравнительный анализ алгоритмов показал, что алгоритму нахождения искомого элемента в массиве полным перебором в большинстве случаев требуется большее количество сравнений, чем алгоритму двоичного поиска.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Поиск в массиве [Электронный ресурс]. Режим доступа: https://math.spbu.ru/user/nlebedin/2-semester/theory_search.pdf (дата обращения 13.10.24)
2. Целочисленный двоичный поиск [Электронный ресурс]. Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=Целочисленный_двоичный_поиск (дата обращения 13.10.24)
3. Python3 documentation [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/> (дата обращения 13.10.24)
4. Библиотека Matplotlib [Электронный ресурс]. Режим доступа: <https://matplotlib.org/> (дата обращения 13.10.24)