



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»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7
По курсу: "Анализ алгоритмов"

Студент _____ Федоров Виктор _____

Группа _____ ИУ7-53Б _____

Название предприятия _____ МГТУ им. Н. Э. Баумана, каф. ИУ7 _____

Тема _____ Поиск в словаре. _____

Преподаватель:	_____	Волкова Л.Л.
	подпись, дата	Фамилия, И. О.
Преподаватель:	_____	Строганов Ю.В.
	подпись, дата	Фамилия, И. О.

Оглавление

Введение	3
1 Аналитическая часть	4
1.1 Описание словаря	4
1.2 Теоретические данные	4
1.3 Поиск полным перебором	4
1.4 Бинарный поиск	5
1.5 Частотный анализ	5
1.6 Вывод	6
2 Конструкторская часть	7
2.1 Разработка алгоритмов	7
2.2 Вывод	7
3 Технологическая часть	11
3.1 Выбор ЯП	11
3.2 Сведения о модулях программы	11
3.3 Тестирование	13
3.4 Вывод	13
4 Экспериментальная часть	14
4.1 Временные характеристики	14
4.2 Вывод	14
Заключение	17
Список литературы	18

Введение

С появлением словарей появилась нужда в том, чтобы уметь быстро находить нужное значение по ключу. Со временем стали разрабатывать алгоритмы поиска в словаре. В данной лабораторной работе мы рассмотрим три алгоритма:

1. поиск полным перебором;
2. бинарный поиск;
3. частотный анализ.

Целью данной работы является изучение и реализация трех алгоритмов.

В рамках выполнения работы необходимо решить следующие задачи:

1. изучения трех алгоритмов поиска в словаре;
2. применение изученных основ для реализации поиска значений в словаре по ключу;
3. получения практических навыков;
4. получение замеров времени;
5. описание и обоснование полученных результатов;
6. выбор и обоснование языка программирования, для решения данной задачи.

Глава 1

Аналитическая часть

1.1 Описание словаря

В данной лабораторной работе использовался словарь VR-игр. Ключами в данном словаре были названия игр. Значениями были объекты, которые имели характеристики данной игры, а именно:

- издатель;
- год издания;
- жанр;
- возрастное ограничение;
- цена.

1.2 Теоретические данные

Словари – объекты, записанные парой "ключ-значение". Отличным примером словаря является толковый словарь. В данном словаре ключи – это нужное нам слово, а значения – это значение искомого слова. Ключи в словаре должны быть уникальными.

Поиск необходимой информации в словаре – одна из фундаментальных задач программирования.

1.3 Поиск полным перебором

Поиск полным перебором может производиться в неотсортированном словаре. Весь алгоритм сводится к последовательному прохождению по всему словарю и сравнению всех значений с ключом. Данный алгоритм имеет $N + 1$ случаев и требует N сравнений для худшего случая. Это простейший из алгоритмов поиска. Этот алгоритм не очень эффективен, однако он работает на произвольном списке.

1.4 Бинарный поиск

Бинарный поиск базируется на том, что словарь изначально отсортирован, что позволяет сравнивать ключ с средним элементом, и, если, он меньше, то продолжать искать в левой части, таким же методом, иначе в правой.

На рис. 1.1 показан пример бинарного поиска.

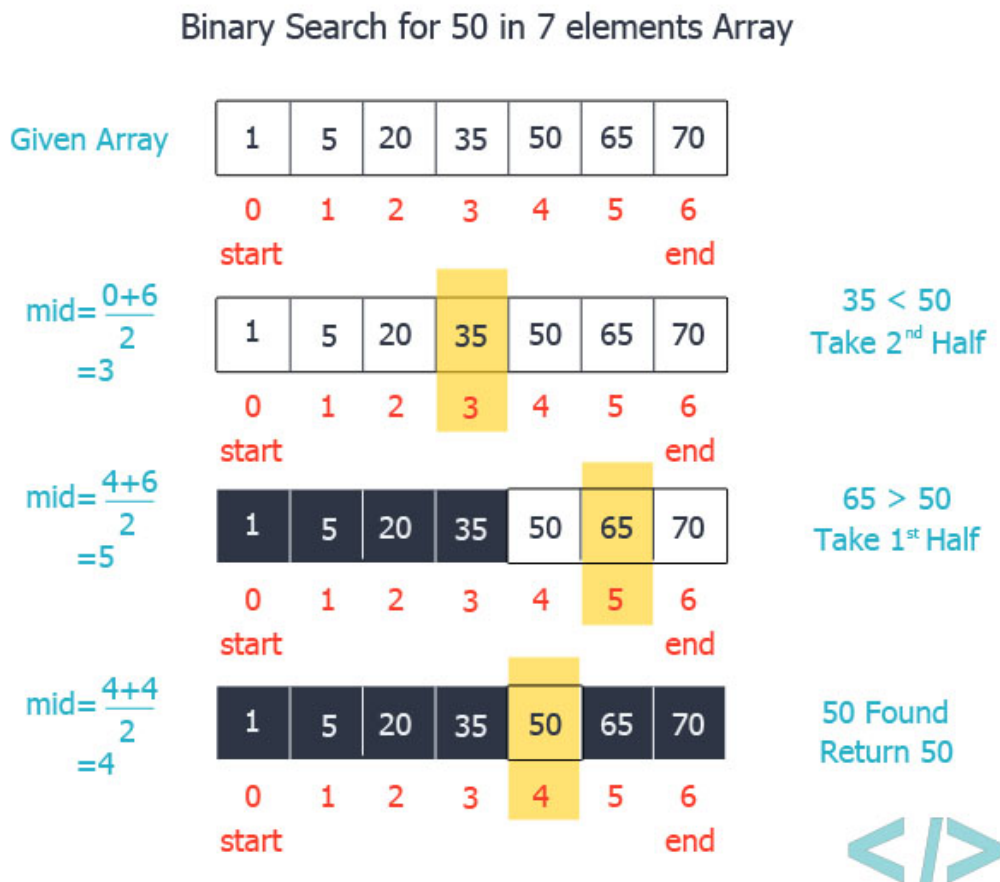


Рис. 1.1: Бинарный поиск

1.5 Частотный анализ

Идея частотного анализа состоит в том, изначально словарь отсортирован по частоте первого символа. Также словарь имеет ключи равные буквам алфавита, а значениями являются словари с исходными данными, которые имеют первую букву схожую с ключом словаря. На деле получается, что это простой толковый словарь, который имеет в заголовке букву, а на странице все слова, которые начинаются на написанную в заголовке букву. Поиск изначально осуществляется по первой букве ключа. Далее, когда найден словарь, в котором нужно искать искомое слово производится поиск полным перебором.

1.6 Вывод

В данном разделе были рассмотрены основополагающие материалы, которые в дальнейшем потребуются при реализации алгоритмов поиска в словаре.

Глава 2

Конструкторская часть

2.1 Разработка алгоритмов

В данном разделе будут рассмотрены схемы.

На рис. 2.1 показана схема алгоритма поиска полным перебором.

На рис. 2.2 показана схема алгоритма бинарного поиска.

На рис. 2.3 показана схема алгоритма частотного анализа.

2.2 Вывод

В данном разделе были рассмотрены схемы алгоритмов.

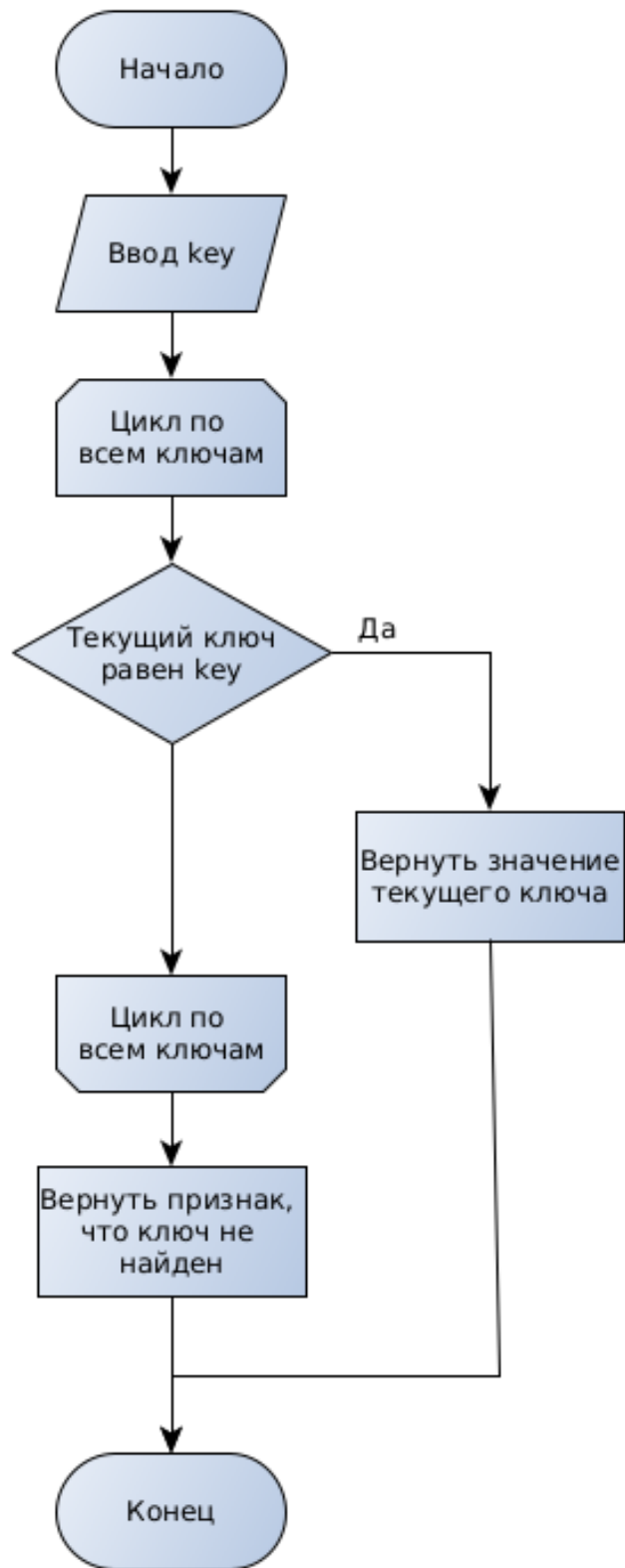


Рис. 2.1: Поиск полным перебором

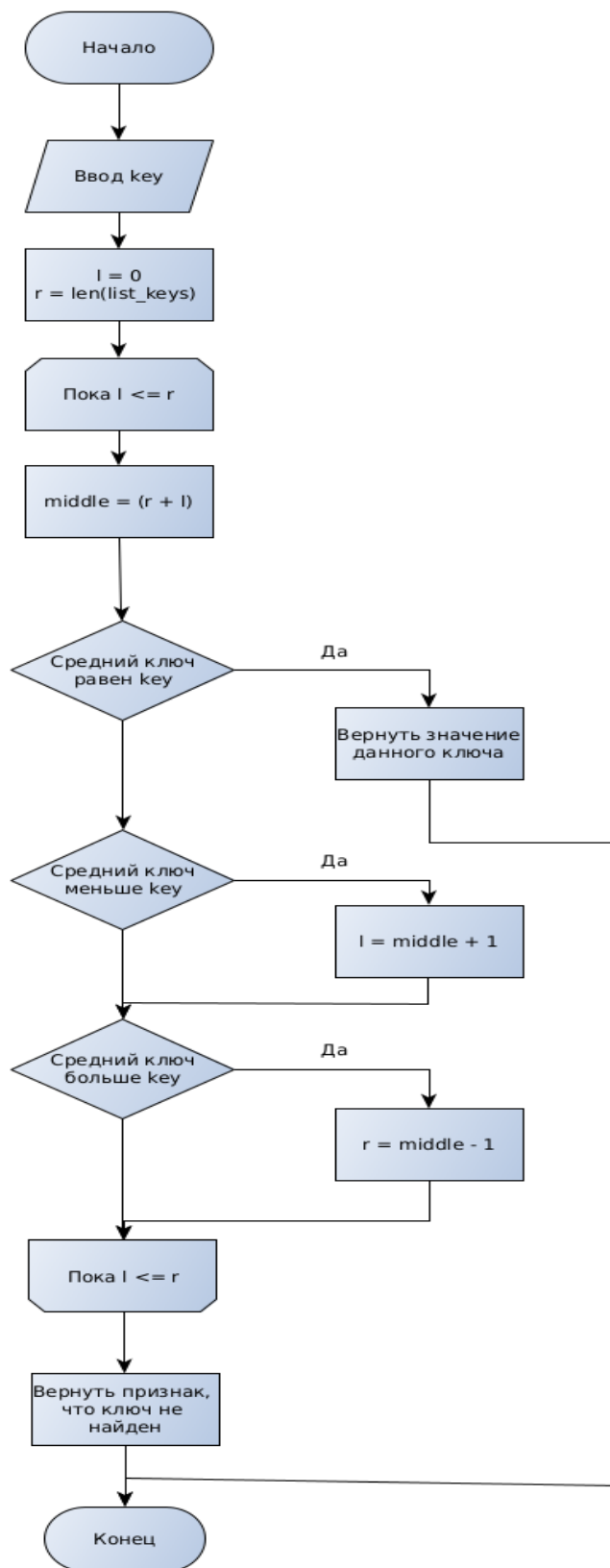


Рис. 2.2: Бинарный поиск

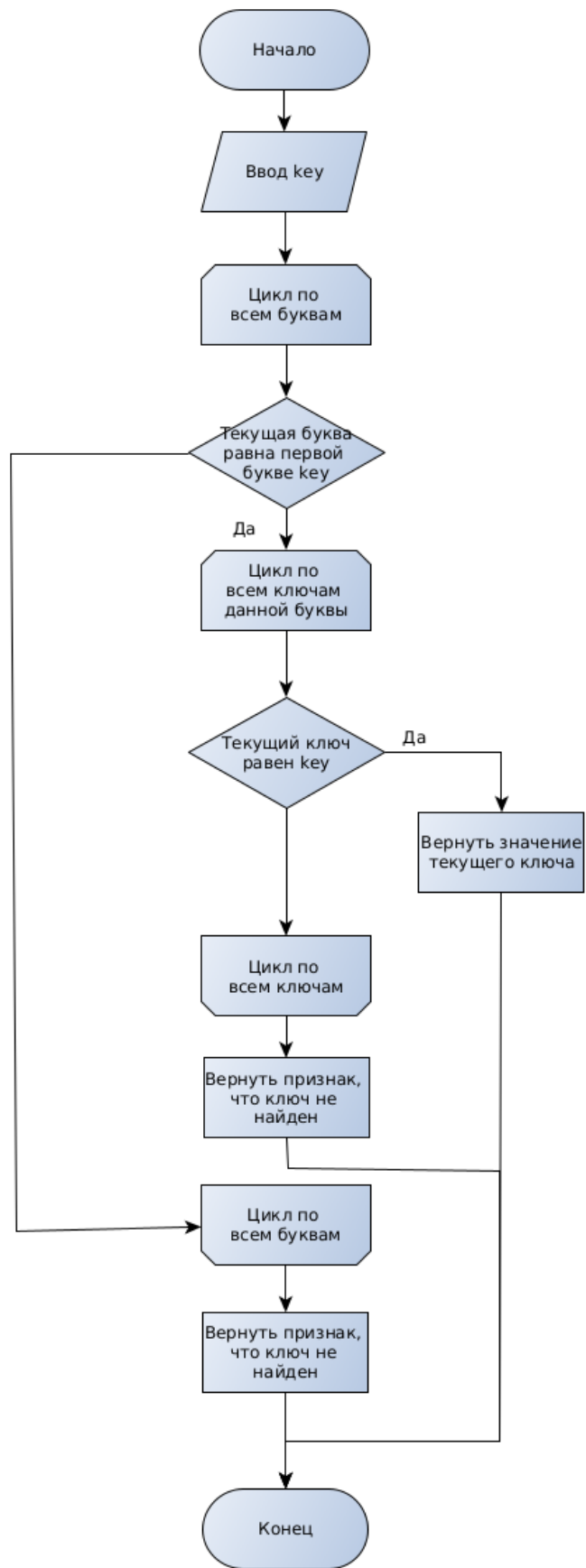


Рис. 2.3: Частотный анализ

Глава 3

Технологическая часть

3.1 Выбор ЯП

В данной лабораторной работе использовался язык программирования python. [4] Язык был выбран, так как я хорошо знаком с ним. В качестве среды разработки использовалась Visual Studio Code [1].

3.2 Сведения о модулях программы

Данная программа разбита на модули.

- main.py - главный файл, содержащий точку входа в программу;
- dictionary.py - файл, содержащий класс dictionary;
- world_info.py - файл, содержащий класс, который описывает информацию о игре.

На листингах 3.1-3.4 представлен основной код программы.

Листинг 3.1: Поиск полным перебором

```
def SequentialSearch(self, key):  
    for elem in self.data:  
        if key == elem:  
            return self.data[elem]  
    return -1
```

Листинг 3.2: Бинарный поиск

```
def BinarySearch(self, key, list_keys):
    l, r = 0, len(list_keys) - 1

    while l <= r:
        middle = (l + (r - l) // 2)

        if list_keys[middle] == key:
            return self.data[elem]

        elif list_keys[middle] < key:
            l = middle + 1

        else:
            r = middle - 1

    return -1
```

Листинг 3.3: Частотный анализ

```
def Search(self, key, new_dict):
    # Loop by letter
    for k in new_dict:
        if key[0] == k:
            for elem in new_dict[k]:
                if elem == key:
                    return new_dict[k][elem]

    return -1

return -1
```

3.3 Тестирование

В данном разделе будет приведена таблица с тестами (таблица 3.1).

Таблица 3.1: Таблица тестов

Входные данные	Пояснение	Результат
A Good Librarian Like a Good Shepherd	Первый элемент	Ответ верный
Championship Manager 2008	Средний элемент	Ответ верный
Patapon 3	Последний элемент	Ответ верный
Adrift	Произвольный элемент	Ответ верный
Adrift 333	Несуществующий элемент	Ответ верный

Все тесты пройдены.

3.4 Вывод

В данном разделе были разобраны листинги рис 3.1-3.4, показывающие работу каждого алгоритма и приведена таблица с тестами (таблица 3.1)..

Глава 4

Экспериментальная часть

В данном разделе будет произведено сравнение трех алгоритмов.

4.1 Временные характеристики

Так как поиск в словаре считается короткой задачей, воспользуемся усреднением массового эксперимента. Для этого сложим результат работы алгоритма n раз ($n \geq 10$), после чего поделим на n . Тем самым получим достаточно точные характеристики времени. Сравнение произведем при $n = 1000$.

На рис. 4.1 приведено сравнение времени выполнения трех алгоритмов для поиска последнего значения. По результатам эксперимента видно, что поиск полным перебором затрачивает больше всего времени, т.к. он последовательно обходит все элементы, в то время, как остальные два алгоритма выполняют поиск значительно быстрее.

На рис. 4.2 произведено аналогичное сравнение, только в качестве искомого ключа взят несуществующий. Аналогично поиск полным перебором затрачивает больше всего времени по вышеописанной причине.

Однако не всегда поиск полным перебором дает худший результат. На рис. 4.3 представлен результат поиска первого значения. Выигрыш алгоритма поиска полным перебором обосновывается тем, что он тратит лишь одно сравнение для того, чтобы найти первый ключ, в то время, когда бинарный поиск затрачивает гораздо больше сравнений. Частотный анализ работает чуть медленнее, так как ему нужно произвести дополнительное сравнение первых букв.

На рис. 4.4 представлен результат поиска произвольного ключа. Алгоритм полного перебора работает медленнее всех.

4.2 Вывод

В данном разделе было произведено сравнение трех алгоритмов. По результатам исследования было доказано, что алгоритм полного перебора в основном работает медленнее всех, за исключением, когда ключ лежит достаточно близко к началу.

```
src [master] ⚡ python3 main.py
Input key: Patapon 3

Value:
founder: Stonecrusher
year_of_issue: 2087
genre: "Artillery game"
age_restrictions: 18
price: 67832

Time sequential search: 0.259133

Time binary search: 0.016861000000000015

Time search: 0.008951999999999996
```

Рис. 4.1: Последний ключ

```
src [master] ⚡ python3 main.py
Input key: Patapon 123

Value:
-1

Time sequential search: 0.275824

Time binary search: 0.016772999999999982

Time search: 0.008717000000000003
```

Рис. 4.2: Несуществующий ключ

```

src [master] ⚡ python3 main.py
Input key: A Good Librarian Like a Good Shepherd

Value:
founder: Malaswyn
year_of_issue: 2074
genre: "Singlplayer"
age_restrictions: 0
price: 77587

Time sequential search: 0.005810000000000003

Time binary search: 0.018867999999999996

Time search: 0.006007999999999997

```

Рис. 4.3: Первый ключ

```

src [master] ⚡ python3 main.py
Input key: Batman

Value:
founder: Burilore
year_of_issue: 2062
genre: "Gamble"
age_restrictions: 6
price: 93379

Time sequential search: 0.072389000000000001

Time binary search: 0.015372000000000001

Time search: 0.014546000000000003

```

Рис. 4.4: Произвольный ключ

Заключение

В данной лабораторной работе были рассмотрены основополагающие материалы которые в дальнейшем потребовались при реализации трех алгоритмов поиска в словаре. Были рассмотрены схемы (рис. 2.1 - 2.3) показывающие работу каждого алгоритма. Также были разобраны листинги (рис 3.1-3.4), была приведена таблица с тестами (таблица 3.1). показывающая корректную работу программы. Был произведен сравнительный анализ (рис. 4.1 - 4.4) и объяснены результаты.

В рамках выполнения работы были изучены три алгоритма поиска в словаре.

Также были решены следующие задачи:

1. применили изученные основы для реализации поиска значений в словаре по ключу;
2. получили практические навыки;
3. получили замеры времени;
4. описали и обосновали полученные результаты;
5. выбрали и обосновали язык программирования, для решения данной задачи.

Литература

- [1] Visual Studio Code [Электронный ресурс], режим доступа: <https://code.visualstudio.com/> (дата обращения: 02.10.2020)
- [2] Windows [Электронный ресурс], режим доступа: <https://www.microsoft.com/ru-ru/windows> (дата обращения: 02.10.2020)
- [3] Linux [Электронный ресурс], режим доступа: <https://www.linux.org.ru/> (дата обращения: 02.10.2020)
- [4] Руководство по языку python [Электронный ресурс], - режим доступа: <https://pythonworld.ru/samouchitel-python> (дата обращения: 26.11.2020)
- [5] Ubuntu 18.04 [Электронный ресурс], режим доступа: <https://releases.ubuntu.com/18.04/> (дата обращения: 02.10.2020)