



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 №6 по курсу "Анализ алгоритмов"

Тема Поиск в словаре

Студент Артюхин Н.П.

Группа ИУ7-51Б

Преподаватели Волкова Л.Л., Строганов Ю.В.

Оглавление

Введение	3
1 Аналитическая часть	4
1.1 Структура данных словарь	4
1.2 Алгоритм приска в словаре	4
2 Конструкторская часть	6
2.1 Разработка алгоритма	6
2.2 Описание используемых типов данных	8
3 Технологическая часть	9
3.1 Требования к программному обеспечению	9
3.2 Выбор средств реализации	9
3.3 Реализация алгоритма	9
3.4 Тестирование	10
4 Исследовательская часть	11
4.1 Пример работы программного обеспечения	11
4.2 Формализация объекта и его признака	12
4.3 Анкетирование респондентов	13
4.4 Построение функции принадлежности термам	14
Заключение	16
Список использованных источников	17

Введение

Целью данной работы является получение навыка поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной.

Для достижения поставленной цели требуется решить следующие задачи:

- 1) разработать алгоритм поиска в словаре объектов;
- 2) описать структуру данных словаря;
- 3) реализовать разработанный алгоритм поиска в словаре;
- 4) формализовать объект по индивидуальному варианту и его признак;
- 5) составить анкету и провести анкетирование респондентов;
- 6) построить функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов, выступающих в роли экспертов;
- 7) описать и обосновать результаты эксперимента.

1 Аналитическая часть

В данном разделе будут описаны словарь как структура данных и алгоритм поиска в словаре.

1.1 Структура данных словарь

Словарь — абстрактный тип данных, позволяющий хранить пары вида «(ключ, значение)» и поддерживающий операции добавления пары, а также поиска и удаления пары по ключу [1]:

- 1) *insert(key, val)*;
- 2) *find(key)*;
- 3) *remove(key)*.

В паре (key, val) : *val* называется значением, ассоциированным с ключом *key*. Здесь *key* — это ключ, а *val* — значение. Семантика и названия вышеупомянутых операций в разных реализациях ассоциативного массива могут отличаться.

Операция поиска *find(key)* возвращает значение, ассоциированное с заданным ключом, или некоторый специальный объект, означающий, что значения, ассоциированного с заданным ключом, нет. Две другие операции ничего не возвращают (только информацию об успешности выполнения операции).

1.2 Алгоритм поиска в словаре

В качестве алгоритма поиска в словаре был выбран алгоритм полного перебора. Алгоритм полного перебора — это алгоритм разрешения математических задач, который можно отнести к классу способов нахождения решения рассмотрением всех возможных вариантов [2]. В случае реализации алгоритма в рамках данной работы будут последовательно перебираться ключи словаря до тех пор, пока не будет найден нужный.

Трудоёмкость алгоритма зависит от того, присутствует ли искомый ключ в словаре, и, если присутствует — насколько он далеко от начала массива ключей.

Пусть в начале своей работы производит n_0 операций, а при сравнении n_1 операций. Лучший случай происходит, когда алгоритм находит нужный элемент при первом сравнении, будет произведено $n_0 + n_1$ операций, худший случай, когда алгоритм находит нужный элемент, перебрав все элементы или если данного ключа нет в массиве, то будет произведено $n_0 + N \cdot n_1$ операций, где N — общее число ключей в словаре. Трудоёмкость в среднем может быть рассчитана как математическое ожидание по формуле (1.1), где Ω — множество всех возможных случаев.

$$\sum_{i \in \Omega} p_i \cdot f_i = n_0 + n_1 \cdot \left(1 + \frac{N}{2} - \frac{1}{N+1} \right) \quad (1.1)$$

Вывод

В данном разделе были описаны словарь как структура данных и алгоритм поиска в словаре.

2 Конструкторская часть

В данном разделе будет представлена схема алгоритма поиска в словаре полным перебором и будут описаны используемые типы данных.

2.1 Разработка алгоритма

На рисунке 2.1 представлена схема алгоритма поиска в словаре полным перебором.

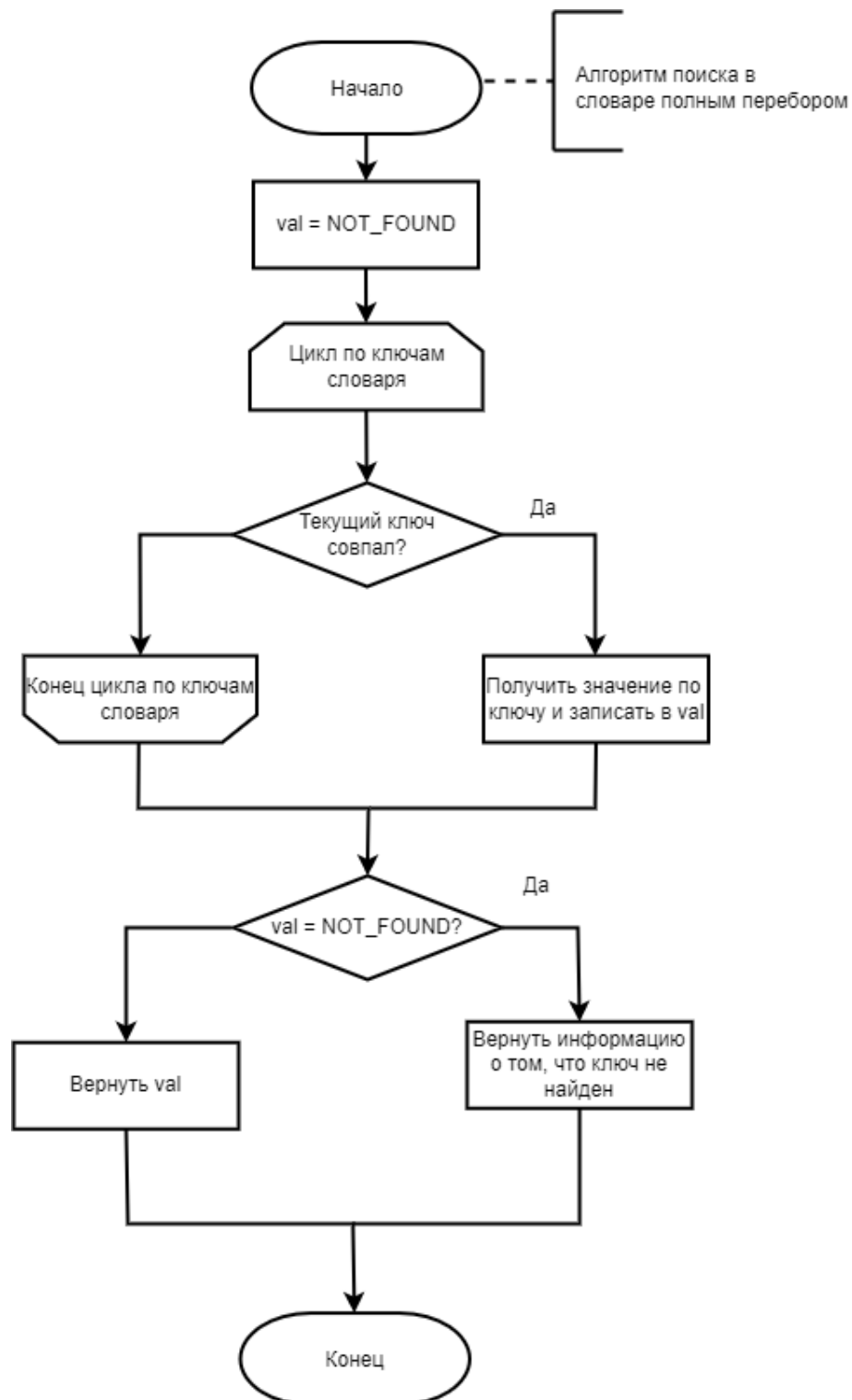


Рисунок 2.1 – Схема алгоритма поиска в словаре полным перебором

2.2 Описание используемых типов данных

При реализации алгоритмов будут использованы следующие типы данных:

- 1) словарь — встроенный тип `dict` [3] в Python [4];
- 2) массив ключей — встроенный тип `list` (список) [5] в Python [4];
- 3) длина массива/словаря — целое число.

Вывод

В данном разделе была построена схема алгоритма поиска в словаре полным перебором и были описаны используемые типы данных.

3 Технологическая часть

В данном разделе будут представлены требования к программному обеспечению, средства реализации, листинги кода и тесты.

3.1 Требования к программному обеспечению

Вход: строка (вопрос), на основании которой будет производиться поиск.

Выход: результат поиска в словаре (футболисты с заданным диапазоном трансферной стоимости).

3.2 Выбор средств реализации

В качестве языка программирования для реализации данной лабораторной работы был выбран язык программирования Python [4]. В данном языке программирования существует встроенный тип данных словарь (dict), также данный язык имеет необходимые библиотеки для построения графиков.

В качестве среды разработки был выбран PyCharm Professional [6]. Данная среда разработки является кросс-платформенной, предоставляет функциональный отладчик, средства для рефакторинга кода и возможность установки необходимых библиотек при необходимости.

3.3 Реализация алгоритма

В листингах 3.1 представлена реализация алгоритма поиска в словаре полным перебором.

Листинг 3.1 – Реализация алгоритма поиска в словаре полным перебором

```
1 def full_search(self, key):  
2     val = NOT_FOUND
```

```

3     keys = list(self.data.keys())
4     for k in keys:
5         if key == k:
6             val = self.data[k]
7                 break
8     return val

```

3.4 Тестирование

В таблице 3.1 приведены функциональные тесты для алгоритма поиска в словаре полным перебором. Все тесты были пройдены успешно.

Таблица 3.1 – Тестирование реализации алгоритма поиска в словаре полным перебором

Входные данные	Ожид. результат	Фактич. результат
очень низкая	$(0, 10]$	$(0, 10]$
высокая	$[111, 150]$	$[111, 150]$
средняя	$[51, 110]$	$[51, 110]$
хорошая	Не найдено	Не найдено
111	Не найдено	Не найдено

Вывод

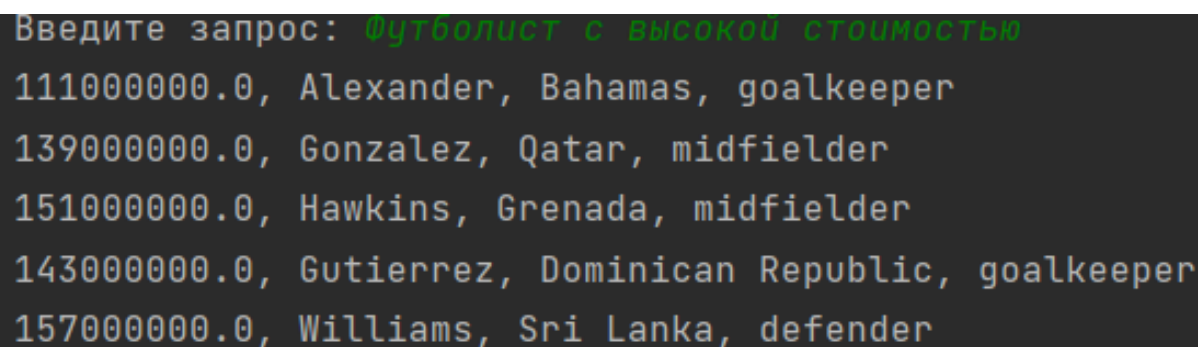
В данном разделе были представлены требования к программному обеспечению и средства реализации, реализован и протестирован алгоритм поиска в словаре полным перебором.

4 Исследовательская часть

В текущем разделе будут представлены пример работы разработанного программного обеспечения, постановка эксперимента и описание полученных результатов.

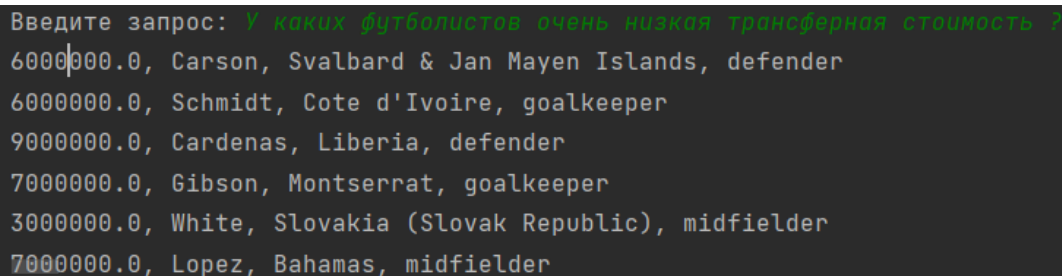
4.1 Пример работы программного обеспечения

На рисунках 4.1 – 4.3 представлены результаты работы программы, на вход программе подается вопрос, если соответствующий терм и объект найдены в вопросе, то выводится информация о футболистах (трансферная стоимость, фамилия, страна рождения, игровая позиция) с заданным диапазоном трансферной стоимости иначе выводится ошибка.



```
Введите запрос: Футболист с высокой стоимостью
111000000.0, Alexander, Bahamas, goalkeeper
139000000.0, Gonzalez, Qatar, midfielder
151000000.0, Hawkins, Grenada, midfielder
143000000.0, Gutierrez, Dominican Republic, goalkeeper
157000000.0, Williams, Sri Lanka, defender
```

Рисунок 4.1 – Пример работы программы



```
Введите запрос: У каких футболистов очень низкая трансферная стоимость ?
6000000.0, Carson, Svalbard & Jan Mayen Islands, defender
6000000.0, Schmidt, Cote d'Ivoire, goalkeeper
9000000.0, Cardenas, Liberia, defender
7000000.0, Gibson, Montserrat, goalkeeper
3000000.0, White, Slovakia (Slovak Republic), midfielder
7000000.0, Lopez, Bahamas, midfielder
```

Рисунок 4.2 – Пример работы программы

Введите запрос: *Баскетболисты с высокой стоимостью*
В запросе должна идти речь о футболистах!

Рисунок 4.3 – Пример работы программы

4.2 Формализация объекта и его признака

Согласно индивидуальному варианту, формализуем объект «футболист» следующим образом: определим набор данных и признак объекта, на основании которого составим набор термов.

Набор данных:

- 1) фамилия футболиста — строка;
- 2) страна рождения — строка;
- 3) игровая позиция — строка.

Признаком, по которому будет производиться поиск объектов, будет *трансферная стоимость* в миллионах евро — вещественное число.

Определим следующие термы, соответствующие признаку «трансферная стоимость»:

- 1) «Очень низкая»;
- 2) «Низкая»;
- 3) «Средняя»;
- 4) «Высокая»;
- 5) «Очень высокая».

Также введём для данной задачи интервал оцениваемой величины (трансферной стоимости) P :

$$P \in (0, 200) \quad (4.1)$$

4.3 Анкетирование респондентов

Было проведено анкетирование следующих респондентов:

- 1) Косарев Алексей, группа ИУ7-51Б — Респондент 1;
- 2) Котляров Никита, группа ИУ7-51Б — Респондент 2;
- 3) Корниенко Клим, группа ИУ7-51Б — Респондент 3;
- 4) Кормановский Михаил, группа ИУ7-51Б — Респондент 4;
- 5) Никулина Анна, группа ИУ7-51Б — Респондент 5;
- 6) Бурлаков Илья, группа ИУ7-51Б — Респондент 6;
- 7) Кузьмин Серафим, группа ИУ7-51Б — Респондент 7.

Респонденты, выступающие в качестве экспертов, для каждого из приведённых выше термов указали соответствующий промежуток из введенного для поставленной задачи интервала оцениваемой величины.

Результаты анкетирования перечисленных респондентов представлены в таблице 4.1. В данной таблице Р. — сокращение от «Респондент», Т. 1 – 5 — термы, соответствующие обозначенным в п. 4.2 термам.

Таблица 4.1 – Тестирование реализации алгоритма поиска в словаре полным перебором

	Т. 1	Т. 2	Т. 3	Т. 4	Т. 5
Р. 1	(0, 10)	[10, 60)	[60, 100)	[100, 160)	[160, 200)
Р. 2	(0, 5)	[5, 50)	[50, 120)	[120, 160)	[160, 200)
Р. 3	(0, 8)	[8, 40)	[40, 110)	[110, 150)	[150, 200)
Р. 4	(0, 15)	[15, 70)	[70, 130)	[130, 185)	[185, 200)
Р. 5	(0, 10)	[10, 40)	[40, 100)	[100, 170)	[170, 200)
Р. 6	(0, 1)	[1, 15)	[15, 60)	[60, 100)	[100, 200)
Р. 7	(0, 10)	[10, 70)	[70, 100)	[100, 150)	[150, 200)

4.4 Построение функции принадлежности термам

Для каждого целого значения, кратного 10 из P для каждого термина из перечисленных будет найдено количество респондентов, согласно которым данное значение из P удовлетворяет сопоставляемому терму. Это значение, поделенное на количество респондентов — значение функции μ для термина в точке. Графики функций принадлежности числовых значений трансферной стоимости термам, представлен на рисунке 4.4.

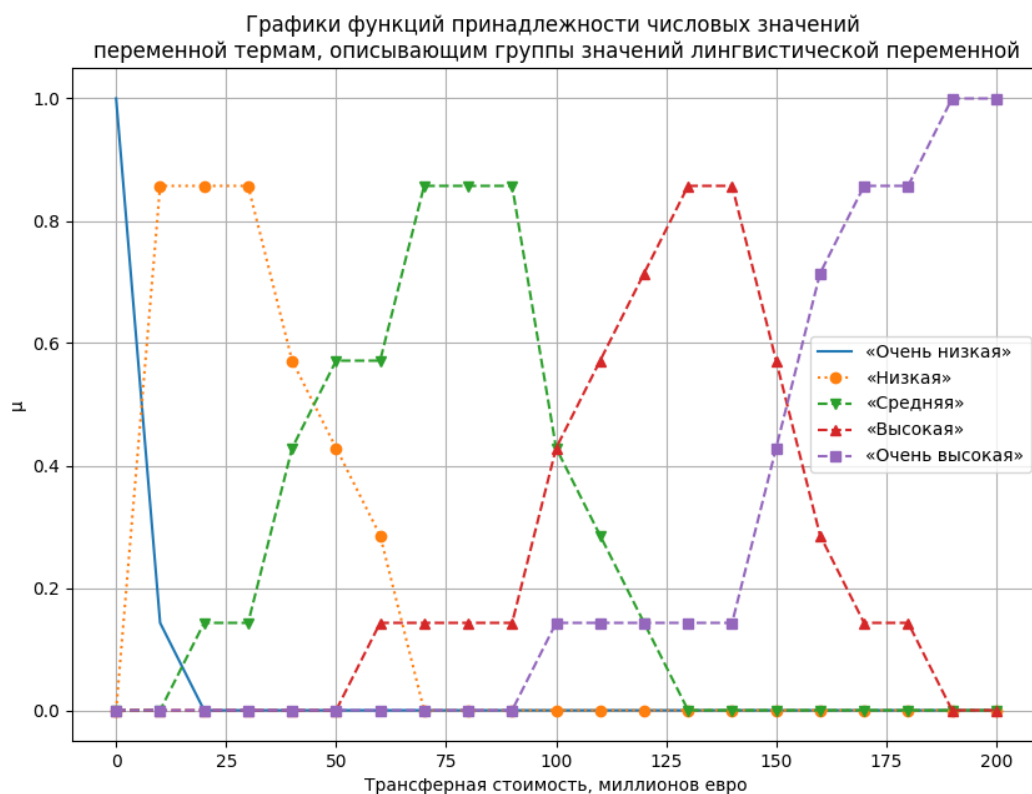


Рисунок 4.4 – Графики функций принадлежности числовых значений переменной термам, описывающим группы значений лингвистической переменной

В соответствии с полученным графиком будем считать трансферную стоимость футболистов:

- 1) очень низкой, если ее значение лежит в промежутке $(0; 10]$ миллионов евро;

- 2) низкой, если ее значение лежит в промежутке $(10; 50]$ миллионов евро;
- 3) средней, если ее значение лежит в промежутке $(50; 110]$ миллионов евро;
- 4) высокой, если ее значение лежит в промежутке $(110; 150]$ миллионов евро;
- 5) очень высокой, если ее значение лежит в промежутке $(150; 200)$ миллионов евро.

Вывод

В данном разделе были представлены примеры работы разработанного программного обеспечения, постановка эксперимента и описание его результатов.

Заключение

В результате выполнения лабораторной работы цель достигнута: получен навык поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной.

В ходе выполнения данной работы были решены все задачи:

- 1) разработан алгоритм поиска в словаре объектов;
- 2) реализован разработанный алгоритм поиска в словаре;
- 3) формализован объект по индивидуальному варианту и его признак;
- 4) составлена анкета и проведено анкетирование респондентов;
- 5) реализованы разработанные алгоритмы для решения задачи коммивояжера;
- 6) построена функция принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов, выступающих в роли экспертов;
- 7) описаны и обоснованы результаты эксперимента.

В соответствии с проведенным экспериментом будем считать трансферную стоимость футболистов:

- 1) очень низкой, если ее значение лежит в промежутке $(0; 10]$ миллионов евро;
- 2) низкой, если ее значение лежит в промежутке $(10; 50]$ миллионов евро;
- 3) средней, если ее значение лежит в промежутке $(50; 110]$ миллионов евро;
- 4) высокой, если ее значение лежит в промежутке $(110; 150]$ миллионов евро;
- 5) очень высокой, если ее значение лежит в промежутке $(150; 200)$ миллионов евро.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. С. Шапошникова. Словари — Лаборатория линуксоида [Электронный ресурс]. 2022. URL: Режим доступа: <https://younglinux.info/python/dictionary> (дата обращения: 19.12.2022).
2. Алгоритм полного перебора [Электронный ресурс]. Режим доступа: https://spravochnick.ru/informatika/algoritmizaciya/algoritm_polnogo_perebora/ (дата обращения: 24.12.2022).
3. dict Python [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/library/2to3.html?highlight=dict#to3fixer-dict> (дата обращения: 04.09.2022).
4. Лутц, Марк. Изучаем Python, том 1, 5-е изд. Пер. с англ. — СПб.: ООО “Диалектика”, 2019 — С. 832.
5. list Python [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/library/pdb.html?highlight=list#pdbcommand-list> (дата обращения: 04.09.2022).
6. Узнайте все о PyCharm [Электронный ресурс]. Режим доступа: <https://www.jetbrains.com/ru-ru/pycharm/learn/> (дата обращения: 20.09.2022).