

**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 № 6 по дисциплине «Анализ алгоритмов»

Тема Поиск в словаре

Студент Калашников С.Д.

Группа ИУ7-53Б

Преподаватель Волкова Л.Л., Строганов Ю.В.

Москва, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитическая часть	4
1.1 Словарь как структура данных	4
1.2 Алгоритм полного перебора	5
1.3 Вывод	5
2 Конструкторская часть	6
2.1 Описание используемых типов данных	6
2.2 Описание алгоритмов	6
3 Технологическая часть	8
3.1 Требования к программе	8
3.2 Средства реализации	8
3.3 Сведения о модулях программы	8
3.4 Реализация алгоритмов	9
4 Исследовательская часть	10
4.1 Формализация объекта и его признака	10
4.2 Анкетирование респондентов	11
4.3 Функция принадлежности термам	11
4.4 Вывод	12
ЗАКЛЮЧЕНИЕ	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	15

ВВЕДЕНИЕ

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

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) формализовать объект по варианту и его признак;
- 2) составить анкета для заполнения респондентом;
- 3) провести анкетирование респондентов;
- 4) построить функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов, выступающих в роли экспертов;
- 5) описать алгоритм поиска в словаре объектов;
- 6) описать структуру данных словаря;
- 7) реализовать описанный алгоритм поиска в словаре;
- 8) описать и обосновать результаты в виде отчёта о выполненной лабораторной работе, выполненном как расчётно-пояснительная записка к работе.

1 Аналитическая часть

В данном разделе будут рассмотрены словарь как структура данных и алгоритм полного перебора.

1.1 Словарь как структура данных

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

1) $insert(k, v)$;

2) $find(k)$;

3) $remove(k)$.

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

Операция поиска $find(k)$ возвращает значение, ассоциированное с заданным ключом, или некоторый специальный объект, означающий, что значения, ассоциированного с заданным ключом, нет. Две другие операции ничего не возвращают (за исключением, возможно, информации о том, успешно ли была выполнена данная операция).

Словарь с точки зрения интерфейса удобно рассматривать как обычный массив, в котором в качестве индексов можно использовать не только целые числа, но и значения других типов — например, строки (именно по этой причине словарь также иногда называют «ассоциативным массивом»).

1.2 Алгоритм полного перебора

Алгоритмом полного перебора называют метод решения задачи, при котором по очереди рассматриваются все возможные варианты. 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 на старте алгоритм затрагивает k_0 операций, а при сравнении k_1 операций.

Пусть алгоритм нашёл элемент на первом сравнении (лучший случай), тогда будет затрачено $k_0 + k_1$ операций, на втором — $k_0 + 2 \cdot k_1$, на последнем (худший случай) — $k_0 + N \cdot k_1$. Если ключа нет в массиве ключей, то мы сможем понять это, только перебрав все ключи, таким образом трудоёмкость такого случая равно трудоёмкости случая с ключом на последней позиции. Трудоёмкость в среднем может быть рассчитана как математическое ожидание по формуле (1.1), где Ω — множество всех возможных случаев.

$$\sum_{i \in \Omega} p_i \cdot f_i = k_0 + k_1 \cdot \left(1 + \frac{N}{2} - \frac{1}{N+1} \right) \quad (1.1)$$

1.3 Вывод

В данном разделе были рассмотрены словарь как структура данных и алгоритм полного перебора.

2 Конструкторская часть

2.1 Описание используемых типов данных

При реализации алгоритмов будут использованы следующие типы данных:

- 1) словарь;
- 2) список ключей;
- 3) длина массива/словаря — целое число типа `int`.

2.2 Описание алгоритмов

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

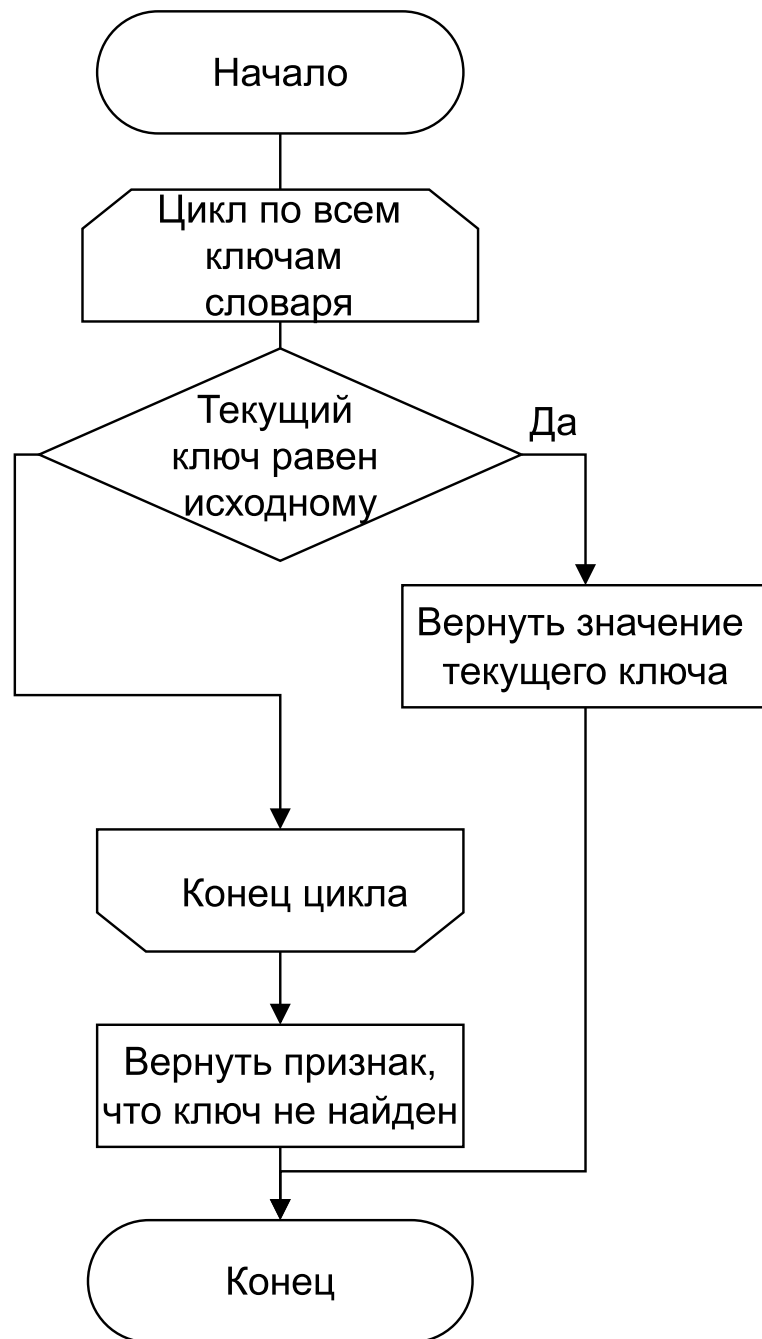


Рис. 2.1 – Схема алгоритма полного перебора

Вывод

В данном разделе были описаны используемые структуры и приведены схемы алгоритмов.

3 Технологическая часть

В данном разделе будут рассмотрены средства реализации, а также представлены листинги алгоритмов.

3.1 Требования к программе

К разрабатываемой в данной работе программе предъявляется ряд требований:

- 1) на вход будет подаваться строка, на основании которой производится поиск;
- 2) на выходе — результат поиска в словаре;
- 3) программа не должна аварийно завершаться при отсутствии ключа в словаре.

3.2 Средства реализации

В данной работе для реализации был выбран язык программирования *python*.

3.3 Сведения о модулях программы

Программа состоит из следующих модулей:

- *main.py* — основной файл программы;
- *utils.py* — файл, содержащий служебные алгоритмы;
- *constants.py* — файл, содержащий константы программы;

3.4 Реализация алгоритмов

В листинге 3.1 представлена реализация алгоритма полного перебора.

Листинг 3.1 — Реализация алгоритма полного перебора

```
1 def full_comb_search(self, key):  
2     k = 0  
3     keys = list(self.data.keys())  
4     for elem in keys:  
5         if key == elem:  
6             return self.data[elem]  
7     return -1
```

Вывод

В данном разделе были рассмотрены требования к программе и листинги используемых алгоритмов.

4 Исследовательская часть

4.1 Формализация объекта и его признака

Согласно согласованному варианту, формализуем объект «время в пути от дома до вуза» следующим образом: определим числовой признак объекта, на основании которого составим набор термов.

Согласно варианту, признаком, по которому будет производиться поиск объектов, будет являться время в минутах — целое число.

Определим следующие термы, соответствующие признаку «время»:

- 1) «Очень долго»;
- 2) «Долго»;
- 3) «Нормально»;
- 4) «Близко»;
- 5) «Очень близко».

Также введём числовое множество множество, описывающее термы:

$$M = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120\} \quad (4.1)$$

4.2 Анкетирование респондентов

Было проведено анкетирование следующих респондентов:

- 1) Светличная Алина — Респондент 1;
- 2) Марченко Владислав — Респондент 2;
- 3) Царев Антон — Респондент 3;
- 4) Лагутин Даниил — Респондент 4.

Респонденты, выступающие в качестве экспертов, для каждого из приведённых выше термов указали соответствующий промежуток, элементами которого являются числа из введённого для поставленной задачи множества оцениваемой величины.

Результаты анкетирования перечисленных респондентов продемонстрированы в таблице 4.1. В данной таблице Респ. — сокращение от «Респондент».

Таблица 4.1 — Результаты анкетирования

Терм	Респ. 1	Респ. 2	Респ. 3	Респ. 4
1	90–120	80–120	100–120	80–120
2	50–90	60–80	80–100	60–80
3	30–50	40–60	50–80	40–60
4	20–30	30–40	30–50	20–40
5	10–20	10–30	10–30	10–20

4.3 Функция принадлежности термам

Графики функций принадлежности числовых значений временным термам, приведён на рис. 4.1.

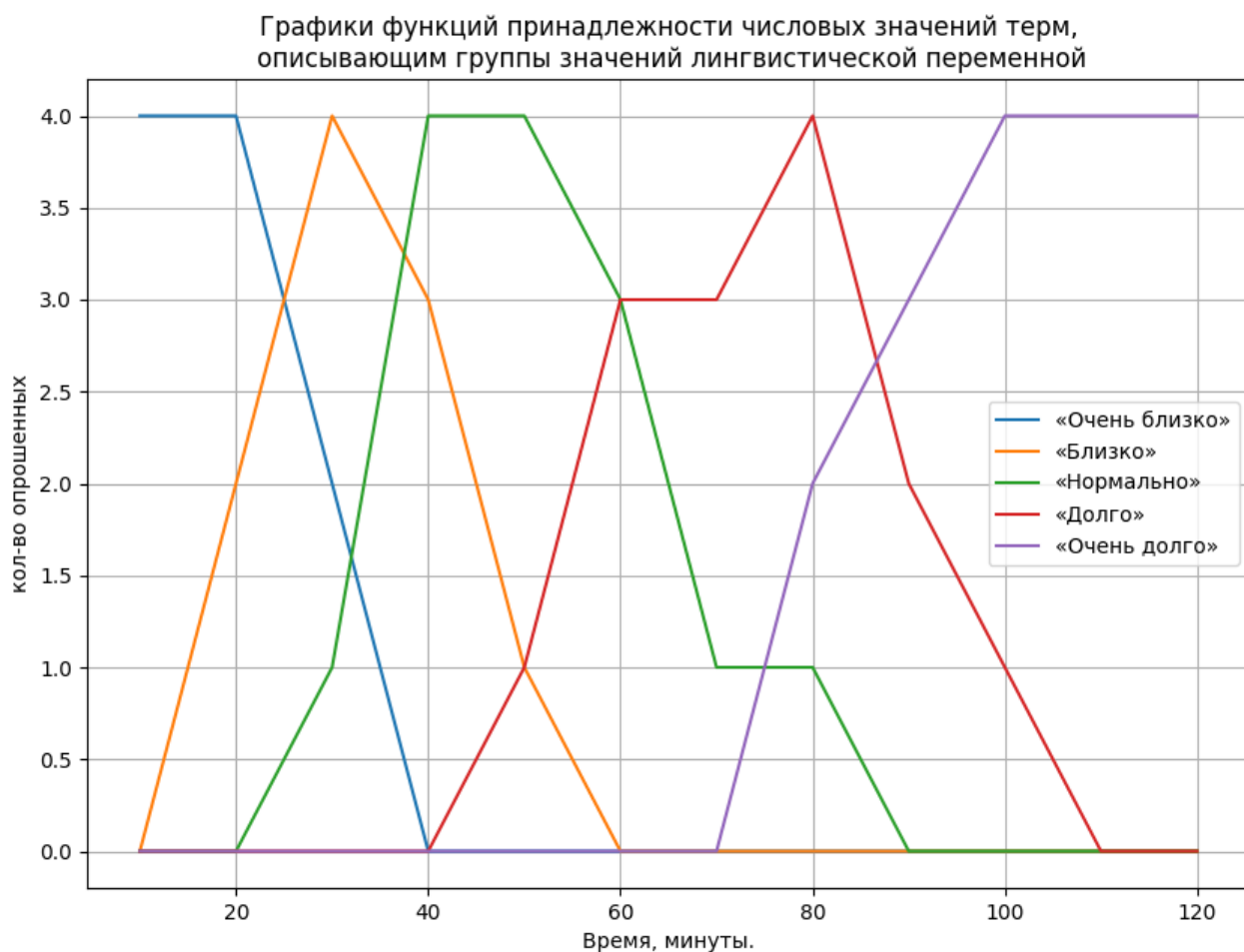


Рис. 4.1 – Зависимость словесной оценки респондентов от количества минут, потраченных на дорогу от дома до вуза

4.4 Вывод

По полученным результатам исследования можно сделать вывод, что в основном время, проведенное в дороге распределяется следующим образом:

- 1) очень долго — от 50 до 120 минут;
- 2) долго — от 60 до 110 минут;
- 3) нормально — от 30 до 80 минут;

4) близко — от 20 до 40 минут;

5) очень близко — от 10 до 30 минут.

ЗАКЛЮЧЕНИЕ

Поставленная цель достигнута: получен навык поиска по словарю при ограничении на значение признака, заданного при помощи лингвистической переменной.

В ходе выполнения лабораторной работы были решены все задачи:

- 1) формализован объект по варианту и его признак;
- 2) составлена анкета для заполнения респондентом;
- 3) проведено анкетирование респондентов;
- 4) построена функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов, выступающих в роли экспертов;
- 5) описан алгоритм поиска в словаре объектов;
- 6) описана структуру данных словаря;
- 7) реализован описанный алгоритм поиска в словаре;
- 8) полученные результаты описаны в виде отчёта о выполненной лабораторной работе, выполненном как расчётно-пояснительная записка к работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Windows 11, version 22H2 [Эл. ресурс]*. Режим доступа: <https://clck.ru/32NCXx> (дата обращения: 14.10.2022).
2. *Процессор Intel® Core™ i7 [Эл. ресурс]*. Режим доступа: <https://clck.ru/yeQa8> (дата обращения: 14.10.2022).