



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ _____ «09.03.04 Программная инженерия»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Название: _____ Поиск по словарю

Дисциплина: _____ Анализ алгоритмов

Студент	<u>ИУ7-56Б</u>	_____	<u>Ковель А.Д.</u>
	Группа	Подпись, дата	И. О. Фамилия

Преподаватель	_____	<u>Волкова Л.Л.</u>
---------------	-------	---------------------

Преподаватель	_____	<u>Строганов Ю.В.</u>
	Подпись, дата	И. О. Фамилия

Москва, 2022 г.

Оглавление

Введение	3
1 Аналитическая часть	4
1.1 Словарь как структура данных	4
1.2 Алгоритм полного перебора	4
2 Конструкторский часть	6
2.1 Описание используемых типов данных	6
2.2 Разработка алгоритмов	6
2.2.1 Разработка простого DBSCAN	6
3 Технологический часть	8
3.1 Требования к программе	8
3.2 Средства реализации	8
3.3 Реализации алгоритма	8
3.4 Тестовые данные	9
4 Исследовательская часть	10
4.1 Технические характеристики	10
4.2 Формализация объекта и его признака	10
4.3 Демонстрация работы программы	11
4.3.1 Построение функции принадлежности термам	12
Заключение	13
Список использованных источников	14

Введение

В процессе развития компьютерных систем количество обрабатываемых данных увеличивалось, вследствие чего множество операций над наборами данных стали выполняться очень долго, поскольку чаще всего это был обычный перебор. Это вызвало необходимость создать новые алгоритмы, которые решают поставленную задачу на порядок быстрее стандартного решения прямого обхода. В том числе это касается и словарей, в которых одной из основных операций является операция поиска.

Цель лабораторной работы — описание поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной.

Задачи данной лабораторной:

- описать объект по варианту и его признак;
- провести анкетирование респондентов;
- описать алгоритм поиска в словаре объектов;
- описать структуру данных словаря;
- реализовать описанный алгоритм поиска в словаре.

1 Аналитическая часть

В этом разделе представляется описание словаря как структура данных и алгоритм полного перебора.

1.1 Словарь как структура данных

Словарь [1] — 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 [2] называют метод решения задачи, при котором по очереди рассматриваются все возможные варианты. В слу-

чае 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.

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

Вывод

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

2 Конструкторский часть

В данном разделе представлены описание используемых типов данных, а также схемы алгоритмов поиска в словаре.

2.1 Описание используемых типов данных

При реализации алгоритмов будут использованы следующие типы данных:

- 1) словарь — встроенный тип `dict` [3] в Python[4] будет использован в созданном классе `Dictionary`;
- 2) массив ключей — встроенный тип `list` [5] в Python[4];
- 3) длина массива/словаря — целое число `int`.

2.2 Разработка алгоритмов

2.2.1 Разработка простого DBSCAN

На рисунке 2.1 приведена схема поиска в словаре полным перебором.

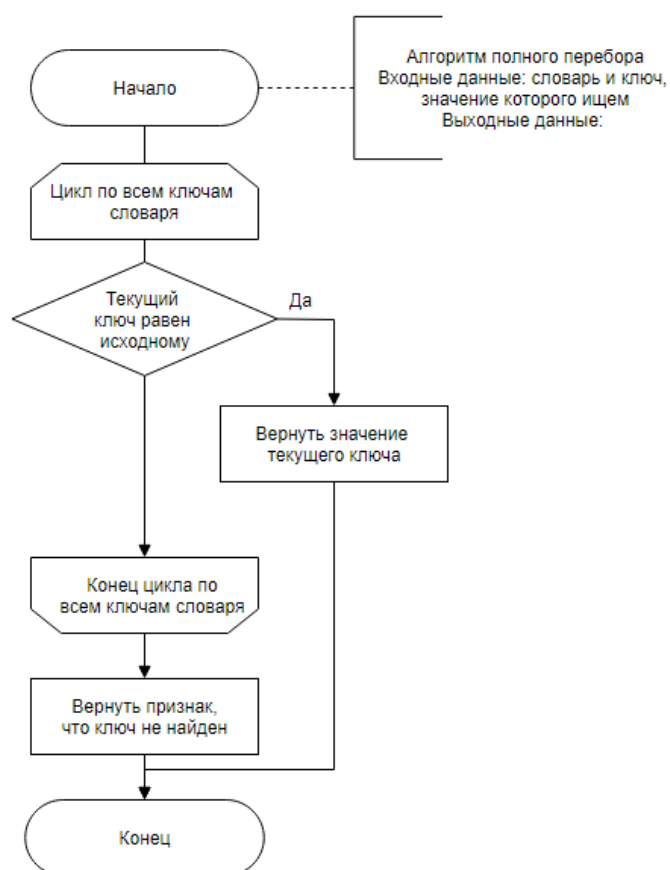


Рисунок 2.1 – Схема плотностного алгоритма DBSCAN

Вывод

Была разработана схема алгоритма, необходимая для решения задачи.

3 Технологический часть

В данном разделе приведены требования к программному обеспечению, средства реализации и листинга кода.

3.1 Требования к программе

Программное обеспечение должно удовлетворять следующим требованиям:

- на вход подается строка;
- на выходе — результат поиска в словаре;
- программа не должна аварийно завершаться при отсутствии в словаре.

3.2 Средства реализации

Для реализации ПО был выбран язык программирования *Python* [4]. В данном языке есть все требующиеся инструменты для данной лабораторной работы. В качестве среды разработки была выбрана среда VS Code [7], запуск происходил через команду `python main.py`.

3.3 Реализации алгоритма

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

Листинг 3.1 – Реализация алгоритма поиска полным перебором

```
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
```

3.4 Тестовые данные

В данном разделе приведена таблица с тестами (таблица 3.1). Применена методология черного ящика. Тесты для всех алгоритмов пройдены *успешно*.

Таблица 3.1 – Таблица тестов

Входные данные	Пояснение	Результат
средний	Средний элемент	Ответ верный
легкий	Первый элемент	Ответ верный
не употребительный	Последний элемент	Ответ верный
балтика	Несуществующий элемент	Ответ верный (-1)
1477	Несуществующий элемент	Ответ верный (-1)

Вывод

Написано и протестировано программное обеспечение для решения поставленной задачи.

4 Исследовательская часть

4.1 Технические характеристики

Тестирование выполнялось на устройстве со следующими техническими характеристиками:

- Операционная система Pop!_OS 22.04 LTS [8] Linux [9];
- Оперативная память 16 Гбайт;
- Процессор AMD® Ryzen 7 2700 eight-core processor × 16 [10].

Во время тестирования устройство было подключено к блоку питания и не нагружено никакими приложениями, кроме встроенных приложений и системой тестирования.

4.2 Формализация объекта и его признака

Согласно варианту, признаком, по которому будет производиться поиск объектов, будет являться *содержание спирта* в градусах — целое число.

Определим следующие термы, соответствующие признаку «содержание спирта»:

- 1) «Легкий»;
- 2) «Средний»;
- 3) «Интересный»;
- 4) «Высокий»;
- 5) «Не употребительный».

4.3 Демонстрация работы программы

Результат работы программы, в которой выводится время работы алгоритма представлено на рисунке 4.1.

```
Озерная с не употребительный(5)
(base) alexander@pop-os:~/Documents/bmstu/ics7-aa/lab-06/app$ python main.py
Введите терм:
не употребительный интересный спирт
Разбитые токены: ['не', 'употребительный', 'интересный', 'спирт']
Найденные термы:
> не употребительный(5),
['не употребительный', 'интересный']
[5, 3]
> интересный(3)
Подпадающие под условия участки:
Балтика 3 с интересный(3)
Балтика 6 с не употребительный(5)
Балтика 5 с интересный(3)
Клинское с интересный(3)
Деревенька с не употребительный(5)
Крушовиче с интересный(3)
Апсент с не употребительный(5)
Мотор с интересный(3)
Озерная с не употребительный(5)
```

Рисунок 4.1 – Пример работы программы

4.3.1 Построение функции принадлежности термам

Построим графики функций принадлежности числовых значений переменной термам, описывающим группы значений лингвистической переменной.

Для этого для каждого значения из количества спирта для каждого термина из перечисленных найдём количество респондентов, согласно которым значение удовлетворяет сопоставляемому терму. Данное значение поделим на количество респондентов — это и будет значением функции μ для термина в точке. Графики функций принадлежности числовых значений роста термам, приведён на рисунке 4.2.

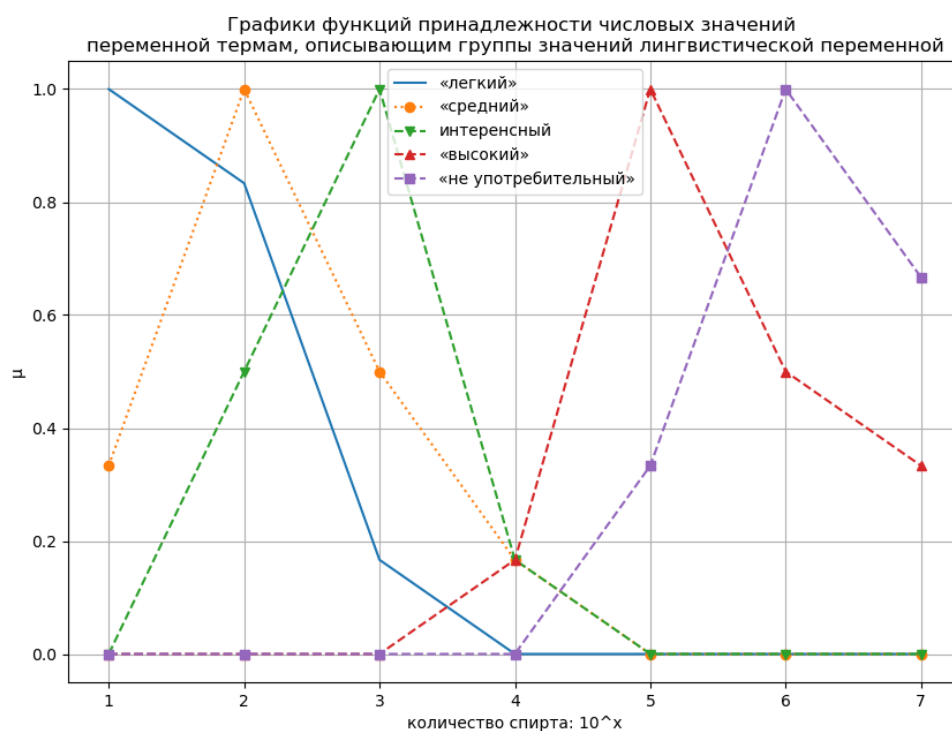


Рисунок 4.2 – Графики функций принадлежности числовых значений переменной термам

Заключение

В ходе выполнения лабораторной работы были решены следующие задачи:

- описан объект по варианту и его признак;
- проведено анкетирование респондентов;
- описан алгоритм поиска в словаре объектов;
- описана структура данных словаря;
- реализован описанный алгоритм поиска в словаре.

Поставленная цель достигнута: описан поиск по словарю.

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] Н. Нильсон. Искусственный интеллект. Методы поиска решений. М.: Мир, 1973. с. 273.
- [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] Welcome to Python [Электронный ресурс]. Режим доступа: <https://www.python.org> (дата обращения: 04.09.2022).
- [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] Golang Документация[Электронный ресурс]. Режим доступа: <https://go.dev/doc/> (дата обращения: 24.09.2022).
- [7] Vscode Документация[Электронный ресурс]. Режим доступа: <https://ru.bmstu.wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C> (дата обращения: 02.11.2022).
- [8] Pop OS 22.04 LTS [Электронный ресурс]. Режим доступа: <https://pop.system76.com> (дата обращения: 04.09.2022).
- [9] Linux – Документация [Электронный ресурс]. Режим доступа: <https://docs.kernel.org> (дата обращения: 24.09.2022).
- [10] Процессор AMD® Ryzen 7 2700 eight-core processor × 16 [Электронный ресурс]. Режим доступа: <https://www.amd.com/en/products/cpu/amd-ryzen-7-2700> (дата обращения: 04.09.2022).