



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»

Отчёт по лабораторной работе №6 по дисциплине «Анализ алгоритмов»

Тема: Поиск в словаре

Студент: Княжев А. В.

Группа: ИУ7-52Б

Оценка (баллы): _____

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

Москва — 2022 г.

Оглавление

Введение	3
1. Аналитическая часть	5
1.1. Формализация объекта и его признака	5
1.2. Двоичный поиск в словаре	5
2. Конструкторская часть	6
2.1. Описание структуры данных словаря	6
2.2. Разработка алгоритма двоичного поиска в словаре	6
3. Технологическая часть	8
3.1. Требования к ПО	8
3.2. Средства реализации	8
3.3. Реализация структуры словаря	8
3.4. Реализации алгоритмов	9
3.5. Тестирование	10
4. Экспериментальная часть	11
4.1. Анкета для респондентов	11
4.2. Результаты анкетирования	11
4.3. Функция принадлежности	14
4.4. Соответствие признаков и диапазонов значений	15
Заключение	16
Список использованных источников	17

Введение

При работе с большими объемами текстовой информации часто возникает необходимость организации ее хранения, удобной для поиска. Одной из структур данных, используемой при работе с текстом, является словарь, который представляет собой набор элементов типа «ключ-значение».

Цель работы

Получение навыка поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной.

Задачи работы

- 1) формализовать объект и его признак;
- 2) составить анкету для её заполнения респондентом;
- 3) провести анкетирование респондентов;
- 4) построить функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов;
- 5) описать 3 – 5 типовых вопросов на русском языке, имеющих целью запрос на поиск в словаре;
- 6) описать алгоритм поиска в словаре объектов, удовлетворяющих ограничению, заданному в вопросе на ограниченном естественном языке;
- 7) описать структуру данных словаря, хранящего наименования объектов согласно варианту и числовое значение признака объекта;
- 8) реализовать описанный алгоритм поиска в словаре;
- 9) привести примеры запросов пользователя и сформированной реализацией алгоритма поиска выборки объектов из словаря, используя составленные респондентами вопросы;

10) дать заключение о применимости предложенного алгоритма и о его ограничениях.

1. Аналитическая часть

В данном разделе рассмотрена формализация объекта и его признака, а также описан алгоритм поиска в словаре.

1.1. Формализация объекта и его признака

Словарь в данном случае представлен в виде списка объектов с признаками. Объектом является название пива. Признаком является крепость пива [1].

При анализе признака в таком словаре осуществляется поиск диапазона допустимых значений для крепости пива.

1.2. Двоичный поиск в словаре

Двоичный поиск — алгоритм поиска, работающий на отсортированных данных, и позволяющий осуществлять поиск элемента с трудоемкостью $O(\log N)$.

Идея алгоритма заключается в том, что исходный словарь разделяется на две половины, и проверяется срединный элемент, если он больше искомого, то в правой половине словаря поиск уже не имеет смысла, так как элементы в правой части больше срединного. Соответственно, имеет смысл дальнейшее рассмотрение только левой части.

Аналогичные выводы можно сделать для ситуации, когда срединный элемент меньше искомого. Тогда следует рассматривать только правую часть словаря.

2. Конструкторская часть

В данном разделе представлены описание структуры данных словаря, и также алгоритм двоичного поиска в словаре.

2.1. Описание структуры данных словаря

Словарь представляет собой список элементов. Каждый элемент списка представляет собой запись, содержащую два поля:

- *strength* — крепость пива, число с плавающей точкой;
- *data* — информация о пиве, например, название, фирма и так далее.

2.2. Разработка алгоритма двоичного поиска в словаре

Пусть

- D — словарь;
- n — количество элементов в словаре;
- v — искомое значение;
- *flag* — если значение «истина», то вычисляется индекс ближайшего элемента, большего данного, иначе — меньшего.

1. Вычислить левую границу обрабатываемого диапазона:

$$l = 0. \tag{2.1}$$

2. Вычислить правую границу обрабатываемого диапазона:

$$r = n. \tag{2.2}$$

3. Пока $l \leq r$

- (а) Вычислить индекс середины диапазона:

$$m = \frac{l + r}{2}. \tag{2.3}$$

- (b) Если $v < D_m$, то $r = m - 1$.
 - (c) Иначе, если $v > D_m$, то $l = m + 1$.
 - (d) Иначе вернуть m .
4. Если $(D_l > D_r) = flag$, то вернуть l .
5. Вернуть r .

3. Технологическая часть

В данном разделе представлены реализации структуры словаря и алгоритма двоичного поиска в словаре. Кроме того, указаны требования к ПО и средства реализации.

3.1. Требования к ПО

- программа позволяет вводить имя файла, содержащего информацию о наборах слов, с помощью аргументов командной строки;
- программа аварийно завершается в случае ошибок, выводя сообщение о соответствующей ошибке;
- программа принимает на вход строку запроса в свободном формате вида: «выдай очень крепкое пиво»;
- программа предлагает исправление опечаток в запросе.

3.2. Средства реализации

Для реализации данной работы выбран язык программирования Go, так как он содержит необходимые для тестирования библиотеки, а также обладает достаточными инструментами для реализации ПО, удовлетворяющего требованиям данной работы [4].

3.3. Реализация структуры словаря

В листингах 3.1 – 3.2 представлена реализация структуры данных «словарь».

Листинг 3.1 — Реализация структуры словаря

```
type Beer struct {  
    Name string  
    Strength float64  
}
```


Листинг 3.2 — Реализация структуры словаря (продолжение листинга 3.1)

```
type Dictionary struct {  
    Elements []Beer  
}
```

3.4. Реализации алгоритмов

В листинге 3.3 представлена реализация алгоритма двоичного поиска по словарю.

Листинг 3.3 — Реализация алгоритма

```
func (d *Dictionary) search(v float64, flag bool) int {  
    if v > d.Elements[len(d.Elements) - 1].Strength {  
        return len(d.Elements) - 1  
    } else if v < d.Elements[0].Strength {  
        return 0  
    }  
    l, r := 0, len(d.Elements) - 1  
    for l <= r {  
        m := (l + r) / 2  
        if v > d.Elements[m].Strength {  
            l = m + 1  
        } else if v < d.Elements[m].Strength {  
            r = m - 1  
        } else {  
            return m  
        }  
    }  
    if d.Elements[l].Strength > d.Elements[r].Strength == flag {  
        return l  
    }  
    return r  
}
```

3.5. Тестирование

Тестирование проводилось по методологии чёрного ящика. **Тесты пройдены успешно.**

В таблице 3.1 представлены тестовые данные для реализаций выполнения запросов с использованием поиска по словарю.

Таблица 3.1 — Тестовые данные для реализаций выполнения запросов с использованием поиска по словарю

№	Запрос	Результат
1	«какое пиво для детей»	Возможно, вы имели в виду «какое пиво для детей». 1. Балтика № 0 «Безалкогольное» - 0.5% 2. Балтика № 7 «Безалкогольное» - 0.5% 3. Пиво Primator Chipper - 2.0% 4. Hoegaarden, Radler Lemon Lime - 2.0% 5. Балтика № 1 «Лёгкое» - 3.0% 6. Балтика № 2 «Светлое» - 3.5% 7. Жигулевское - 3.5% 8. Балтика № 3 «Классическое» - 3.8%
2	«крепкое пиво»	1. Балтика № 9 «Крепкое» - 8.5% 2. Охота крепкое - 8.5%
3	«квашенные гвозди»	не понятно, о чем запрос
4	«льмкльедлк»	не найдено слово

4. Экспериментальная часть

Для формирования системы запросов о крепости пива, возникла необходимость провести опрос среди респондентов и построить функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной.

В данном разделе приведена анкета, отправленная респондентам. Также представлены результаты анкетирования и статистической обработки мнений респондентов.

4.1. Анкета для респондентов

В таблице 4.1 представлена таблица-анкета, предоставленная респондентам.

Таблица 4.1 — Анкета, предоставленная респондентам

Ид. респ.	Терм i	Крепость пива ξ , градусов						
		0	1	3	5	7	9	11
1	безалкогольное							
	некрепкое							
	слабое							
	нормальное							
	крепкое							
	очень крепкое							

4.2. Результаты анкетирования

В таблице 4.2 представлено соответствие идентификаторов анкетлируемых их фамилиям. Респондентами выступали сокомандники в рамках практикума по курсу «Архитектура ЭВМ».

В таблице 4.3 представлены результаты анкетирования.

Таблица 4.2 — Соответствие идентификатора респондента и респондента

Ид.	Респондент
1	Карпова Е. О.
2	Глютов И. А.
3	Ляпина Н. В.
4	Аскарян К. А.
5	Обревская В. В.

Таблица 4.3 — Результаты анкетирования респондентов

Ид. респ.	Терм i	Крепость пива ξ , градусов						
		0	1	3	5	7	9	11
1	безалкогольное	1	1	0	0	0	0	0
	некрепкое	0	0	1	0	0	0	0
	слабое	0	0	0	1	0	0	0
	нормальное	0	0	0	0	1	1	0
	крепкое	0	0	0	0	0	0	0
	очень крепкое	0	0	0	0	0	0	1
2	безалкогольное	1	0	0	0	0	0	0
	некрепкое	0	1	0	0	0	0	0
	слабое	0	0	0	0	0	0	0
	нормальное	0	0	1	1	0	0	0
	крепкое	0	0	0	0	1	1	0
	очень крепкое	0	0	0	0	0	0	1
3	безалкогольное	1	0	0	0	0	0	0
	некрепкое	0	1	1	0	0	0	0
	слабое	0	0	0	0	0	0	0
	нормальное	0	0	0	1	0	0	0
	крепкое	0	0	0	0	0	0	0
	очень крепкое	0	0	0	0	1	1	1
4	безалкогольное	1	1	0	0	0	0	0
	некрепкое	0	0	1	0	0	0	0
	слабое	0	0	0	1	0	0	0
	нормальное	0	0	0	0	1	0	0
	крепкое	0	0	0	0	0	1	0
	очень крепкое	0	0	0	0	0	0	1
5	безалкогольное	1	0	0	0	0	0	0
	некрепкое	0	1	0	0	0	0	0
	слабое	0	0	1	1	0	0	0
	нормальное	0	0	0	0	1	0	0
	крепкое	0	0	0	0	0	1	0
	очень крепкое	0	0	0	0	0	0	1

4.3. Функция принадлежности

В таблице 4.4 представлены значения функции принадлежности термам числовых значений признака, описываемого лингвистической переменной.

Таблица 4.4 — Таблица значений функции принадлежности

безалкогольное	$\mu_j(\xi)$	5	5	2	0	0	0	0
	$F_j(\xi)$	1.0	0.4	0.0	0.0	0.0	0.0	0.0
некрепкое	$\mu_j(\xi)$	0	3	3	0	0	0	0
	$F_j(\xi)$	0.0	0.6	0.6	0.0	0.0	0.0	0.0
слабое	$\mu_j(\xi)$	0	0	1	4	0	0	0
	$F_j(\xi)$	0.0	0.0	0.2	0.6	0.0	0.0	0.0
нормальное	$\mu_j(\xi)$	0	0	1	2	3	1	0
	$F_j(\xi)$	0.0	0.0	0.0	0.2	0.4	0.6	0.2
крепкое	$\mu_j(\xi)$	0	0	0	0	1	3	1
	$F_j(\xi)$	0.0	0.0	0.0	0.0	0.2	0.6	0.2
очень крепкое	$\mu_j(\xi)$	0	0	0	0	1	1	4
	$F_j(\xi)$	0.0	0.0	0.0	0.0	0.2	0.2	0.8

В таблице используются следующие обозначения:

— $\mu_j(\xi) = \sum_{k=1}^N a_{ji}^k$;

— $F_j(\xi) = \frac{S_j}{N}$,

— N — количество респондентов.

На рис. 4.1 изображена функциональная зависимость $F_j(\xi)$.

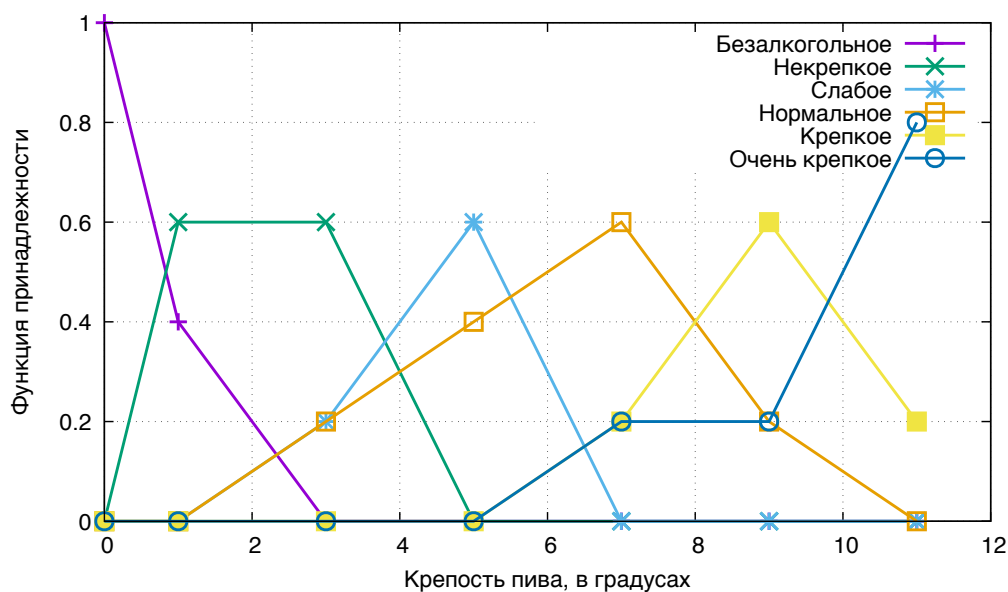


Рисунок 4.1 — Функциональная зависимость

4.4. Соответствие признаков и диапазонов значений

В таблице 4.5 представлено соответствие между термами и диапазонами значений для крепости пива в градусах.

Таблица 4.5 — Соответствие между термами и диапазонами значений для крепости пива в градусах

Признак	Диапазон
безалкогольное	[0.00; 0.83]
некрепкое	[0.83; 3.80]
слабое	[3.80; 5.50]
нормальное	[5.50; 8.00]
крепкое	[8.00; 9.80]
очень крепкое	[9.80; 11.00]

Заключение

Алгоритм двоичного поиска имеет асимптотику $O(\log N)$, то есть является более эффективным алгоритмом, чем, например линейный поиск. Но двоичный поиск требует отсортированности данных. Так как в рамках задачи предполагается упорядочивание данных, то данный алгоритм применим в данной задаче.

Цель работы была достигнута: был получен навык поиска по словарю при ограничении на значение признака, заданном при помощи логической переменной. Были выполнены все задачи:

- формализован объект и его признак;
- составлена анкеты для заполнения респондентами;
- проведено анкетирование респондентов;
- построена функция принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов;
- описаны типовые вопросы на русском языке, имеющие целью запрос на поиск в словаре;
- описаны алгоритмы поиска в словаре объектов, удовлетворяющих ограничению, заданному в вопросе на ограниченном естественном языке;
- описана структура данных словаря, хранящего наименования объектов согласно варианту и числовое значения признака объекта;
- реализован данный алгоритм поиска в словаре;
- приведены примеры запросов пользователя и сформированный реализацией алгоритма поиска выборки объекта из словаря, используя составленные респондентами вопросы;
- дано заключение о применимости предложенного алгоритма и его ограничениях.

Список использованных источников

1. Антониади А. Ю., Ковалева Е. Г. Методы получения пива повышенной крепости //Актуальные вопросы органической химии и биотехнологии. – 2020. – С. 639-640.
2. Федотов А. М. и др. Модель определения нормальной формы слова для казахского языка //Вестник Новосибирского государственного университета. Серия: Информационные технологии. – 2015. – Т. 13. – №. 1. – С. 107-116.
3. Петровский В. И., Бондарев В. Н. Модуль анализа естественно-языкового текста //Интеллектуальные системы, управление и мехатроника-2016. – 2016. – С. 368-372.
4. Документация по языку программирования *Go* [Электронный ресурс]. Режим доступа: <https://go.dev/doc> (дата обращения: 07.10.2022).
5. Документация по пакетам языка программирования *Go* [Электронный ресурс]. Режим доступа: <https://pkg.go.dev> (дата обращения: 07.10.2022).