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"

Тема Поиск в словаре

Студент Цветков И.А.

Группа ИУ7-53Б

Оценка (баллы)

Преподаватель Волкова Л. Л.

Москва — 2021 г.

Содержание

Введение

В процессе развития компьютерных систем количество данных стало достигать огромных размеров, поэтому множество операций стали выполняться очень долго, поскольку чаще всего это был обычный перебор. Это вызвало необходимость создать новые алгоритмы, которые решают поставленную задачу наорядок быстрее стандартоного решения “в лоб”. В том числе это касается и словарей, в которых одной из основных операций является операция поиска.

Целью данной работы является изучение алгоритмов поиска в словаре – полным перебором, бинарным поиском и сегментами. Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучить понятие словаря;
- описать алгоритмы решения задачи поиска в словаре – полный переборбинарный поиск и сегментами;
- привести схемы алгоритмов;
- описать используемые типы и структуры данных;
- описать структуру разрабатываемого программного обеспечения;
- реализовать разработанные алгоритмы;
- провести функциональное тестирование разработанного алгоритма;
- провести сравнительный анализ по времени для реализованного алгоритма;
- подготовить отчет по лабораторной работе.

1 Аналитическая часть

В этом разделе будет представлена информация о словаре, а также об алгоритмах поиска в нем – полным перебором, бинарном и сегемнтами.

1.1 Словарь

Словарь [1] – тип данных, который позволяет хранить пары вида "ключ-значение" – *(k, v)*. Он поддерживает три операции – добавление пары, поиск по ключу, удаление по ключу. В паре *(k, v)* – v это значение, которое ассоциируется с ключом v.

При поиске возвращается значение, которое ассоциируется с данным ключом, или "не найдено", если по данному ключу нет значений.

В данной лабораторной работе:

- ключ – фамилия футболиста;
- значение – информация о нем.

1.2 Алгоритм полного перебора

Полный перебор [2] – метод решения, при котором поочередно перебираются все ключи словаря, пока не будет найден нужный.

Чем дальше искомы ключ от начала словаря, тем выше трудоемкость алгоритм. Так, если на старте алгоритм затрагивает b операций, а при сравнении k операций, то:

- элемент найден на первом сравнении за $b+k$ операций (лучший случай);
- элемент найден на *i-ом* сравнении за $b + i \cdot k$ операций;
- элемент найден на последнем сравнении за $b + N \cdot k$ операций, где N – размер словаря (худший случай);

При этом средняя трудоемкость равна:

$$f = b + k \cdot \left(1 + \frac{N}{2} - \frac{1}{N+1}\right) \tag{1.1}$$

1.3 Бинарный поиск

Бинарный поиск [3] – поиск в заранее отсортированном словаре, который заключается в сравнении со средним элементов, и, если ключ меньше, то продолжать поиск в левой части тем же методом, иначе – в правой части.

Тогда случаи расположены следующим образом (b – кол-во операций алгоритма на старте):

- элемент найден на первом сравнении с средним элементом – трудоемкость $b + \log_2 1$ (лучший случай);
- элемент найден на i-*ом* сравнении – трудоемкость $b + \log_2 i$;
- элемент найден на последнем сравнении – трудоёмкость $b + \log_2 N$, где N – размер словаря (худший случай);

1.4 Поиск с помощью сегментов

Поиск с помощью сегментов [4] – словарь разбивается на части, в каждую из которых попадают все элементы с некоторым общим признаком – одинаковая первая буква, цифра, слово.

Обращение к сегменту равно сумме вероятностей обрщаения к его ключам. Пусть P_i – вероятность обращения к i-ому сегменту, а p_j – вероятность обращения к j-ому элементу i-ого сегмента. Тогда вероятность выбрать нужный сегмент высчитывается так

$$P_i = \sum_j p_j \tag{1.2}$$

Затем ключи в каждом сегменте сортируются, чтобы внутри каждого сегмента можно было произвести бинарный поиск с слонжостью $O(\log_2 k)$, где k – количество ключей в сегменте.

То есть, сначала выбирается нужный сегмент, а затем в нем с помобмощью бинарного поиска ищется нужный ключ.

При этом случаи располагаются так:

- первым выбран верный сегмент, а нужный элемент – серединный (лучший случай);
- нужный сегмент выбран последним, а поиск ключа в данном сегменте – $\log_2 N$, где N - число элементов в сегменте (худший случай);

При этом средняя трудоемкость поиска i-го элемента:

$$\sum_{i \in \Omega} (f_{\text{выбор сегмента i-ого элемента}} + f_{\text{бинарный поиск i-ого элемента}}) \cdot p_i \tag{1.3}$$

1.5 Вывод

В данном разделе было рассмотрено понятие словаря, а также алгоритма поиска в словаре – полным переборм, бинарный и сегментами.

Программа будет получать на вход словарь, а также ключ, по которому нужно будет найти значение в данном словаре одним из трех алгоритмов, который также будет вводиться. Если словарь пуст или какое-то из значений введено неверно, то будет выдано сообщние об ошибке.

Реализуемое ПО дает возможность получить значение по ключу одним из трех алгоритмов поиска в словаре. Также имеется возможность провести тестирование по времени для рассматриваемых алгоритмов.

Список литературы

[1] Словари [Электронный ресурс]. Режим доступа: https://younglinux.info/python/dictionary (дата обращения: 27.11.2021).

[2] Полный перебор [Электронный ресурс]. Режим доступа: http://skud-perm.ru/posts/polnyj-perebor (дата обращения: 27.11.2021).

[3] Бинарный поиск [Электронный ресурс]. Режим доступа: https://prog-cpp.ru/search-binary/ (дата обращения: 27.11.2021).

[4] Нильсон Н. Искусственный интеллект. Методы поиска решений. 1973.