DATA STRUCTURE SEMESTER PROJECT

Jawad Ahmed: BSCS-2019-59
Sharjeel Akram: BSCS-2019-60
Muhammad Anas: BSCS-2019-49

February 1, 2021

DATA STRUCTURE AND ALGORITHM
Sir Malik Jahan
Sir Muhammad Saqib

Table of Contents

Abstract

Search Engine is the most well-known program to search any thing in the big data. Search Engine make the life of other people easy to access the needed and specific data that the user wants. Simple search engine project is implemented in python using several functions and classes having algorithm. Main aim of this project is to develop a search engine which will search in txt file of movies bases on average rating display will display the most popular, least popular and the similar rating movies data. In present trend search engines are used by every for finding required information on the web. Google is one of the mostly used search engine followed by yahoo and bing.

1 Acknowledgements

We acknowledge that this Report for Search Engine In Python is written with contributions of all members of group. We have written algorithms and functions with the help of lab instructor and where we felt difficulty there we took some help from the google.

2 Introduction

2.1 Purpose statement

This report is purposed to make a Search Engine in an optimised way and also compare the performances of different algorithms and functions which are used to make this Search Engine to find movie based on ratings.

2.2 Problem statement

Your task is to build a search engine for the given data (using Python). Data is composed of objects (records). Each object has three attributes: movie ID, average rating and number of votes to review each movie. Your task is populate the data in your search engine based on the average rating.You are required to compare performance of:

1. Hashing (quadratic probing and separate chaining)
2. 2-3-4 tree
3. AVL trees

You can use the search engine to answer questions like:

1. Which one is the most popular movie?
2. Which one is the least popular movie?
3. How many movies have same ratings?

3 Methodology

This part of report is to tell about all the methods which are used to prepare and analyse search engine. We used Python language for source code to run algorithms of search engine. In python we used the concept of oop and data structure having some algorithms implemented using the concept of avl tree, Hashing and also 234 tree.

4 Procedure:

4.1 Reading File:

In order to read txt file in the python we use the function with open and pass the txt file having parameter Read Only ('r') : Open text file for reading.
After reading the file I convert it into python list with function split. And after converting into the txt file into the python list then I slice the python list into 3 different lists Movie Id, Movie Rating and Movie Voting.

4.2 Project With AVL Tree:

AVL is self balancing BST which is efficient in big data as compared to BT or BST.
There are two clases having 15 functions. Almost every function has complexity of O(1) time but the complexity of insert and Search function is O(log(n)) times.

4.2.1 Insert Function:

In our case length of Movie Data is about 9 million and if we find its complexity of insert and Search will be 5.95. And it almost take less than 3 min to insert the whole data in the tree.

4.2.2 Search Function:

As in the Search function recursion is used therefore the complexity of Search Function in worst case become O(n) times otherwise it will be O(log(n)) times.

4.3 Project With Hashing:

Hashing is the process of converting a given key into another value. A hash function is used to generate the new value according to a mathematical algorithm. The result of a hash function is known as a hash value or simply, a hash.
There are three types of hashing to solve collision:
1) Separate Chaining
2)Quadratic Probing
3)Linear Probing
Among these types we implemented two types separate chaining and quadratic probing.
1) Separate Chaining: The complexity of separate chaining is best of all because it insert and search the data in average condition of O(1) times which is faster than any one else.
2) Quadratic Probing: The complexity of quadratic probing is worst then anyone else because the complexity of quadratic probing is O(n) times, the insertion complexity is O(n) times and the probing complexity is O(n) square times.

4.3.1 Insert Function:

In our case length of Movie Data is about 9 million and if we find its complexity of insert, In Separate Chaining it will be seconds. But quadratic probing is opposite to it. It took hour to insert the whole data, because the complexity of probing in quadratic probing is O(n) square times.

4.3.2 Search Function:

As in the hashing Array technique is used and also the method of hashing therefore it took O(1) times to search the data.

4.4 Project With 234 Trees:

A 2-3-4 tree is a balanced search tree having following three types of nodes. 2-node has one key and two child nodes (just like binary search tree node). 3-node has two keys and three child nodes. 4-node has three keys and four child nodes.

Time Complexity Of 234 Tree is following:
1) Search is 2-3-4 Tree takes O(lg N) as the tree is always perfectly balanced.
2) Insertion takes O(lg N) as all splits are local transformations and we don't need multiple passes on the tree.

4.4.1 Insert Function:

In our case length of Movie Data is about 9 million and if we find its complexity of insert and Search will be 3.0 because its complexity is O(log(n)) times. As we do not successfully enter complete data therefore we do not know exactly about the time taken by the 234 trees.

4.4.2 Search Function:

As in the Search function recursion is used therefore the complexity of Search Function in average case will be O(log(n)).

5 Recommendations

The main purpose for this report is to compare the performances of simple search engine with AVL tree, separate chaining, quadratic probing and 234 trees.After successfully insertion of data we noted the complexity and time taken by the insertion as well as the Searching functions for making it clear that which one is best of all the methods.

6 Limitations

This project is done within certain limits.There are multiple issues we faced and also resolved it and some are still existing due to the our system specifications as well as the complexity problem like in quadratic probing when we passed whole data it will stuck and take a lot of time. As the data is inconsistent therefore there is some issues faced in AVL too but later on solved. In 234 tree there is still problems existing in inserting the data.

7 Conclusion

After observing and analysing the data of movies in the AVL Tree, Separate Chaining and Quadratic probing we come to know that hashing is the best of all and more specifically separate chaining is the one which take a lot data with complexity of O(1) times.The AVL take less than 3 min time in inserting the data while quadratic probing took almost 1 hour to insert the data. 234 tree is also good enough to handle the big data but there still existing some issues in implementation.
After carefully comparison of all the Simple Search Engines we came to know that separate chaining is best among all the others and it is also used by Google.