

CALIFORNIA STATE UNIVERSITY, FRESNO
DEPARTMENT OF COMPUTER SCIENCE

Class:	Algorithms & Data Structures			Semester:	Fall 2023
Points		Document author:	Pratham Aggarwal		
		Author's email:	Pratham_aggarwal @mail.fresnostate.edu email		
		Laboratory number:	Lab 7		

1. Statement of Objectives

The objective of this lab is to write a program that counts unique words in a given script using a hash table with linear probing. The scope of this report includes the program's implementation, output analysis, and the difficulties encountered while writing code. The significance of this experiment is in understanding how a hash table works and how it may be used to count unique words in a text file.

2. Experimental Procedure

The program starts by opening and reading the "RomeoAndJuliet.txt" file.

Implementation of a Hash Table: In this case, each unique word serves as the key, and its corresponding count (number of occurrences) serves as the value. The size of the hash table is fixed at 4001, a prime number ideal for storing a large number of unique words.

Processing Each Word: The program then goes through each word in the text file.

- a. Hashing: It computes a hash value for the word to decide where it should be stored in the hash table.
- b. Linear Probing: If a collision occurs, that is, if the calculated index is already taken, the program uses linear probing to locate the next open spot in the hash table.
- c. Updating the Hash Table: After determining the proper index, the program updates the hash table by adding a new entry for the word (with a count of 1) or incrementing the count if the word is already there.

Statistic Collection: While processing the words, it keeps track of the number of updates made (every time a word is added or its count is increased).

After processing the full text file, the program determines the cost per update, providing insight into the efficiency of the hash table and the chosen hash function.

3. Analysis

The output I got is correct since it shows that the program counted 3,684 different words in the "RomeoAndJuliet.txt" text file. The cost per update, on the other hand, quantifies the hash table's efficiency and is determined as the average number of array visits (probes) necessary for each update process.

Screenshot in the end of the report.

4. Encountered Problems

There weren't no major issues that I faced when writing the code. The code was implemented correctly, and the expected results were achieved. However, I struggled with writing code for various areas, like writing hash functions, applying linear probing, and calculating cost per update. For this, I took the help of tutors and some online resources.

5. Conclusions

In this lab, I used a hash table with linear probing to count the number of unique words in a text file. The program identified and counted 3,684 distinct words in "RomeoAndJuliet.txt." This experiment showed the practical application of hash tables for efficiently processing and evaluating textual data. It also demonstrated the significance of choosing a suitable hash function and dealing with probable collisions in a real-world application.

6. References

N/A

[illegible]