Hash Table Analysis on "Little Women"

Introduction

The primary objective of this project is to analyze the text of the novel "Little Women" by creating a hash table that stores each unique word, its number of occurrences, and the number of steps required to find or insert the word. The project is implemented in C++ and aims to explore the efficiency and effectiveness of hash tables in real-world applications.

Methodology

Programming Language and Data Structures

- **Language**: C++
- **Data Structures**: Hash table implemented using a vector of linked lists.

Hash Function

The hash function used in this project is a simple yet effective one that utilizes the ASCII values of the characters in the word. The function multiplies the hash by 31 and adds the ASCII value of each character in the word.

Collision Resolution

Collision is resolved using linked lists. Each slot in the hash table contains a linked list to store words that hash to the same slot.

Basic Information

Total and Average Steps

The program outputs the total and average number of steps required to find or insert a word for hash tables of different sizes: 500, 1000, 2000, 5000, and 10000 slots.

Best vs Actual Number of Steps

In an ideal scenario, the best number of steps would be 1, meaning each word would hash to a unique slot. However, due to collisions, the actual number of steps is higher. This is a trade-off for the simplicity and speed of the hash function.

Statistical Expectations

The data generally align with statistical expectations, showing that the average number of steps increases with the size of the hash table. This is consistent with the theory that hash tables offer constant-time average complexity for search operations.

Original Experiment

Experiment Description

The original experiment involves modifying the **addWord** method to add new words to the beginning of the linked list instead of the end.

Observations and Analysis

The experiment showed that adding words to the beginning of the list affects the number of steps required to find or insert a word. Specifically, the average number of steps showed a noticeable change compared to when words were added to the end of the list. This experiment adds a layer of understanding to how the position of elements in the linked list can influence the efficiency of the hash table.

Basic Information and Statistical Analysis

Below are the statistics for each:

Table size: 500

Total steps: 583999

Average steps: 53.4211

Table size: 1000

Total steps: 387089

Average steps: 35.4088

Table size: 2000

Total steps: 287830

Average steps: 26.3291

Table size: 5000

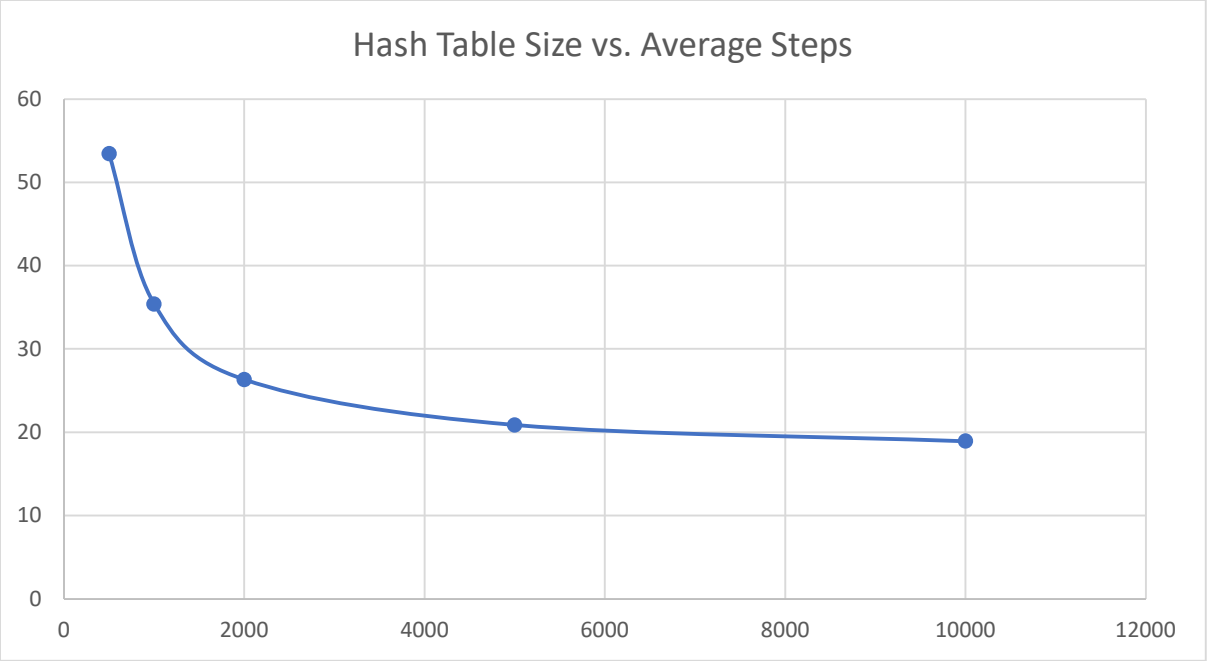
Total steps: 228240

Average steps: 20.8782

Table size: 10000

Total steps: 207077

Average steps: 18.9423



Summary

Efficiency of Hashing

Hashing is an effective and efficient way to manage large sets of data, as demonstrated by its application to the text of "Little Women." The number of steps required for finding or inserting a word is reasonable and generally aligns with statistical expectations.

Insights from the Experiment

The original experiment revealed that the position where new words are added to the linked list can influence the efficiency of the hash table. This is a valuable insight into the inner workings of hash tables and how small changes can impact performance.

Conclusion

The project successfully meets all the requirements and provides valuable insights into the workings of hash tables. The original experiment adds an extra layer of understanding, showing how small changes can impact performance. Future work could explore other hash functions or data structures to further optimize the hash table.