Performance Analysis of Hash Functions

October 10, 2023

1 Introduction

In this analysis, we examine the performance of two hash functions: the Most Significant Bits Method and Cormen's Multiplication Method. We conducted experiments using three datasets, each with 10,000 entries, where each entry has a value (4 alphabet characters) and a random key between 1 and 10,000.

2 Cormen's Multiplication Method

We first explored Cormen's Multiplication Method with hash table sizes of 64 and 256. Our observations are as follows:

- Hash Distribution: At a glance, the hash function appears to distribute keys uniformly across all slots of the hash table.
- Number of slots = 256:
  - Expected number of elements per slot: 39
  - Median number of elements per slot: 46
  - Maximum elements in a slot: 76
  - Minimum elements in a slot: 3
  - Average hashing time: 0.04 seconds
  - Average search time for 100 values: 1.3 seconds
- Number of slots = 64:
  - Expected number of elements per slot: 156
  - Median number of elements per slot: 142
  - Maximum elements in a slot: 198
  - Minimum elements in a slot: 102
  - Average hashing time: 0.04 seconds
  - Average search time for 100 values: 1.6 seconds
- Conclusion: Cormen's Multiplication Method is effective when the range of keys is not known, and this hashing method has a constant upper bound of $O(1)$. It provides a relatively uniform distribution of keys in hash slots.

3 Most Significant Bits Method

We then investigated the Most Significant Bits Method by shifting each key to $12 - p$ bits, where p is the most significant bits of k. We tested with $p = 5$ and hash table size $n = 64$.

- Hash Distribution (p = 5):
  - Expected number of elements per slot: 156
  - Median number of elements per slot: 108

- Maximum elements in a slot: 196
    - Minimum elements in a slot: 70
    - Average hashing time: 0.03 seconds
    - Average search time for 100 values: 1.5 seconds
- Effectiveness: The effectiveness of the Most Significant Bits Method varies depending on the range of the keys, affecting the number of bits used for hashing. The upper bound of this hashing method is approximately $O(p \log_2 k)$.
- If I increase the hash table size to 100, the slots from 64 to 99 are empty.
- Conclusion: The Most Significant Bits Method is effective but dependent on the key range. It provides a reasonable distribution for certain key ranges but may leave some hash slots empty if the range is not well-suited.

4 Conclusion

In conclusion, both hash methods have their strengths and weaknesses.
In my opinion, i would implement Cormen's Multiplication Method when I have a diverse range of key values, and I want a hash function that can evenly distribute keys across hash slots without significant sensitivity to the key range. Consider it for scenarios where the key range is not well-defined or may vary over time, as it provides a constant upper bound and remains efficient under changing conditions.
On the other hand, I would consider using the Most Significant Bits Method when I have a good understanding of the key range and can choose the number of bits accordingly. It can be effective in scenarios where the key range aligns well with the chosen number of bits. Hence, I can optimize memory usage by selecting an appropriate number of bits to minimize hash table size.