

When you are satisfied that your program is correct, write a brief analysis document. The analysis document is 10% of your Assignment 7 grade. Ensure that your analysis document addresses the following.

1. Explain the hashing function you used for BadHashFunc. Be sure to discuss why you expected it to perform badly (i.e., result in many collisions).

The primary function of BadhashFunc is to return the length of the item. The reason that is a bad function since there will be many string items has the same length, and that will create a number of the collisions

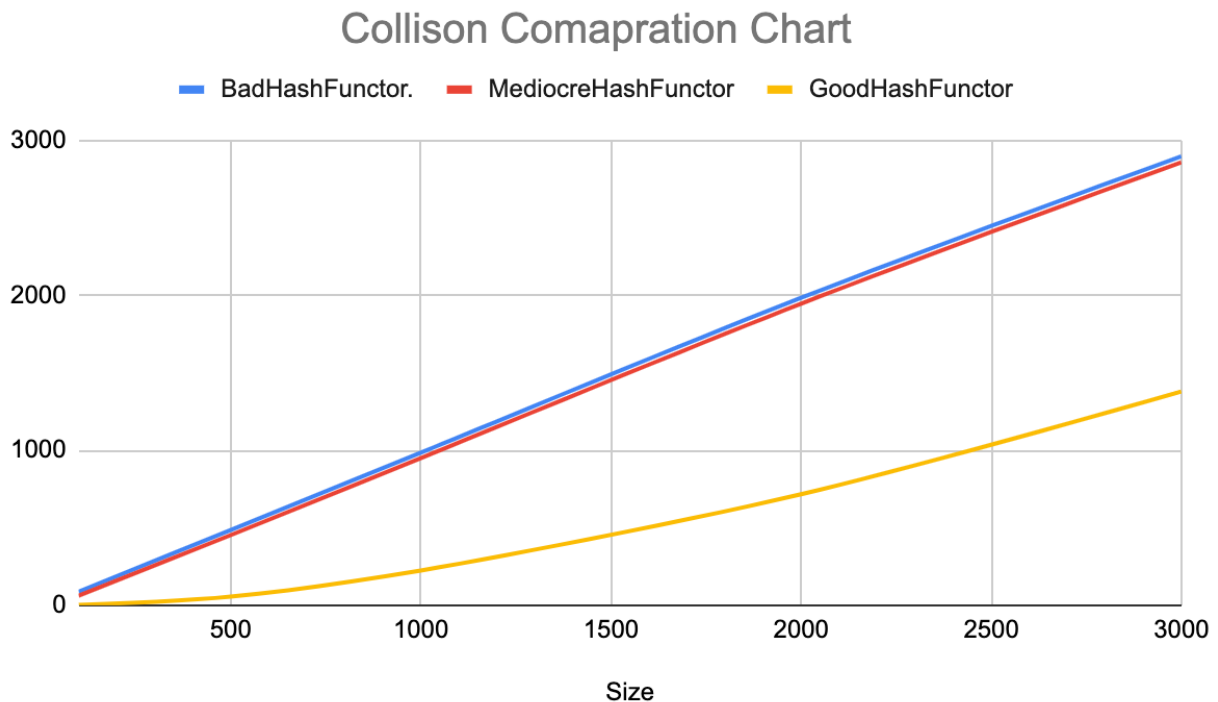
2. Explain the hashing function you used for MediocreHashFunc. Be sure to discuss why you expected it to perform moderately (i.e., result in some collisions).

The major function of the MediocreHashFunc, is to pick up the first character and last character of a string item plus the string item's length. This function is slightly better than the BadHashFunc, which created unique identification. However, vocabulary like "hill" and "hell" would create collisions.

3. Explain the hashing function you used for GoodHashFunc. Be sure to discuss why you expected it to perform well (i.e., result in few or no collisions).

GoodHashFunc has a different way of collecting hashcode compared to MediocreHashFunc and BadHashFunc. It designs to times 23 with each character and sum all of them. Then return an absolute value. Therefore the result will reduce the significant possibility of collision.

4. Design and conduct an experiment to assess the quality and efficiency of each of your three hash functions. Briefly explain the design of your experiment. Plot the results of your experiment. Since the organization of your plot(s) is not specified here, the labels and titles of your plot(s), as well as, your interpretation of the plots is important. A recommendation for this experiment is to create two plots: one that shows the number of collisions incurred by each hash function for a variety of hash table sizes, and one that shows the actual running time required by various operations using each hash function for a variety of hash table sizes.



After multiple testing from 100 to 3000. We can clear sees the GoodHashFuncutor has significant performance compare to other two functor. GoodHashFuncutor create a least collisions.

5. What is the cost of each of your three hash functions (in Big-O notation)? Note that the problem size (N) for your hash functions is the length of the String, and has nothing to do with the hash table itself. Did each of your hash functions perform as you expected (i.e., do they result in the expected number of collisions)?

According to the function and the chart, the BadHashFuncutor and MediocreHashFuncutor grow linearly. Therefore, the cost will be $O(N)$. however, GoodHashFuncutor extends into a curve, and such a graph conforms to the mode of $O(N^2)$. GoodHashFuncutor time complexity will be $O(N^2)$

Upload your solution (.pdf only) through Canvas.