BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE208 (Data Structures and Algorithm Sessional), July 2018 Term

Offline 6: Implementation of Hash Table

---

1 Problem Specification

In this assignment you have to implement ***HashTable*** which will incorporate the following properties:

1. You must have the provision to perform insert, search, delete operations on the hash table. The length of the hash table **N** will be given as input at the beginning.

2. The data will be kept as a key-value pair in the hash table. You need to insert randomly generated 7 character length unique words (these words can be meaningful or meaningless) into the hash table. To do this, you have to implement a random word generator. The words will be used as the key of the hash table. You need to insert those words into the hash table one by one. If your word generator generates duplicate words, you have to discard those duplicate words. The sequence number of generation will be used as the value of the key-value pair.

2 Example

Suppose your word generator has generated the following 5 words:

ancient
puzzled
benefit
ancient
zigzags

The key-value pair will be:

(ancient,1)
(puzzled,2)
(benefit,3)
(zigzags,4)

You have to discard the 2nd ancient since it has already been included in the table.

3 Hash Function

You have to use 2 standard hash functions ($Hash1(k)$ and $Hash2(k)$) of your own/ from any good literature where you must try to avoid collisions as much as possible. We are expecting that 60% key will point to unique hash values (at least 6000 unique hash values for 10000 entries).

4 Collision Resolution

For collision resolution you need to implement three methods:

1. **Chaining Method**
   Place all the elements that hash to the same slot into a linked list. Slot j contains a pointer to the head of the list of all stored elements that hash to j ; if there are no such elements, slot j contains $NULL$.

2. **Double Hashing**
   In case of Double hashing, use the hash function of the form

   $$doubleHash(k, i) = (Hash(k) + i \times auxHash(k)) \ \% \ N$$

   where $Hash(k)$ is one of the hash functions described in Section 3 (Note that you have to run for both the Hash functions; see the Section 5 for more detail) and $auxHash(k)$ is an auxiliary hash function (To avoid complication, you can keep this auxiliary hash function simple). The initial probe goes to position $Table[Hash(k)]$; successive probe positions are offset from previous positions by the amount $auxHash(k)$, modulo N.

3. **Custom Probing**
   In case of Custom Probing, use the hash function of the form

   $$customHash(k, i) = (Hash(k) + C_1 \times i \times auxHash(k) + C_2 \times i^2) \ \% \ N$$

   where C_1 and C_2 are two auxiliary constants which you can choose at your convenient. The other mechanisms should be same as Double Hashing.

5 Report Generation

For report generation you have to generate 10000 seven characters length unique words following exactly the method previously described and insert them into the Hash Table. Mention the number of collisions in a tabular format using both the hash functions Hash1 and Hash2.
From the same set of generated words, pick randomly selected 1000 words and search each word in the hash table and report the average number of hits required to search each of of the words in a tabular format. Also do this for both the Hash functions.

Here number of collisions is the number of occurrences when Hash Function generates already generated hash values for a different word. On the other hand, number of hits required means how many times you have to access the Hash Table in order to search a word.
The report should be generated in the following tabular format

	Hash1		Hash2	
	Number Of Collision	Average Hit	Number Of Collision	Average Hit
Chaining Method				
Double Hashing				
Custom Probing				

Table 1: Report Table

6 Submission deadline

Submission deadline is 1.00 am on 13 January, 2019 (Sunday).

7 Conclusion

For any query please feel free to email at tmadnan10@gmail.com.