Assignment 7  
CISC 610 Data Structures & Algorithms

## By: Syed Mohiuddin Qadri

Answer 1:

Hash table is functionally equivalent to the data structure of a linked list. When many values that are hashed to same slot it would require collision handling. These can be resolved by either open addressing or chaining.

1. Answer 2:
2. 1. Lose-Lose:
   1. Benefits: It’s simple and easy to understand
   2. Drawbacks: Lots of collisions. Not particularly faster than other hashing functions

2. RS Hash:

Benefits: Designed to be simple, Minimizes collisions. Popular C hash algorithm

Drawbacks:

Inefficient when many collisions occur

3. DJB2 Hash:

Benefits: Popular C hash algorithm, Generates specific size of hash i.e. 32-bit

Drawbacks:

Not redefining the multiplier, Collisions can happen

4. SDBM Hash:

Benefits:

Generates unique values, Fast compared DJB2

Drawbacks:

Not good for cryptography

1. Answer 3:
2. 1. Open Addressing: It is a technique of resolving collisions. The data is inserted into the hash table itself. If the slot is free, it inserts the element into that location and if it is not free, it will keep probing until an empty slot is reached. The different ways of Open Addressing are: Linear Probing, Quadratic Probing and Double Hashing

The advantage of this method is additional storage is not required. All data is stored in the same table.

Extra computation is required for open addressing as the next free slot address is calculated at every occurrence of a collision. Information could cluster. Common hashes would lead to increasingly long put / get times. Constant hashing could occur to the same location.

2. Chaining: It is a technique for resolving collisions. In this approach, the hash table is an array of linked lists. That means an index has its own linked list. A linked list is stored, and value is stored at that index when the address is found to be occupied or when collision occur.

The hash table will never fill up as the length of the linked list can be increased. It is easy to implement. Load factor and clustering is not a problem.

Chaining would require additional space to store the linked lists. If the same value is generated by the hash function or if many collisions occur at the same index, it would keep increasing then the length of the linked list.

1. Answer 4:

As the load factor of the hash table grows, it will also increase the number of collisions. This leads to the degradation of the hash table's performance. With chaining, it is acceptable for hash tables, but for open addressing based hash tables performance degradation would be unacceptable. So, when its load factor exceeds the provided threshold, the solution is to resize table.

If the load factor exceeds the threshold, then the hash table needs to increase.

If the load factor too low than the threshold, this would result in wastage of memory. In this case the hash table’s size should be reduced to reduce waste of memory.

A new hash function must be considered when a new table is created. The new hash table size, the computation for copying elements from the previous table and the new hash table resizing. However, creating the new hash table and the performing reinserting the elements to this new table are complex and account for computational load.