

CS 5008

Mid-term

Name: Minjie Shen

1. Explain the importance of Hashmaps in Computer Science. Ensure that you cite credible academic resources.

Hashing is a traditional approach in computer science whereby data structures called hash maps are used to store key-value pairs. Apart from storing key-value pairs, hash maps can be used to solve a wide variety of problems in computer science.

One of the biggest advantages of using hash maps is that they provide a constant time of $O(1)$ for search, insert, and delete operations. This makes them more suitable for problems related to finding duplicates, finding the frequency of items, and finding distinct elements.

2. How good is the provided hash function -- are we really getting constant time operations with our hashmap?

If the hash function is good enough, we may place all elements evenly throughout the buckets we have. In the best case, we can reach constant time for search, insert, and delete operations. However, if the hash function is not well designed, (for example, we might put all elements in one bucket, and all other buckets are empty), we will get $O(n)$ time operations in the worst case.

3. What is one other way you could implement the hash function? Anything creative is acceptable in this answer.

I implemented a new way for the hash function as following:

```

int stringHashNew(char *myKey, int numberOfBuckets){
    int hash = 0;
    for (int i = 0; i < numberOfBuckets; i++){
        hash = 37 * hash + myKey[i];
        hash %= numberOfBuckets;
        if (hash < 0){
            hash += numberOfBuckets;
        }
    }
    return hash;
}

```

We use the ASCII value of the character and add them to a number, then mod this number with the bucket size we have to get the has number.

4. If I have to resize the hashmap to add more buckets, what is the Big-Oh complexity?

For resize the hashmap, the Big-Oh complexity for this operation is $O(N)$, and N stands for how many elements the hashmap has now. For Rehash, make a new array of double the previous size and make it the new bucket array. Then traverse to each element in the old bucket Array and call the insert() (which takes $O(1)$ time) for each so as to insert it into the new larger bucket array.

5. What is "open addressing" in regards to hash maps and hash tables?

Like separate chaining, open addressing is a method for handling collisions. In Open Addressing, all elements are stored in the hash table itself. So at any point, the size of the table must be greater than or equal to the total number of keys.

Citation:

1. Citation for question 1: <https://betterprogramming.pub/the-importance-of-hash-maps-831884307806#:~:text=Hash%20maps%20can%20solve%20a%20whole%20bunch%20of%20different%20problems&text=Hashing%20is%20a%20traditional%20approach,of%20problems%20in%20computer%20science>.
2. Citation for question 4: <https://www.geeksforgeeks.org/load-factor-and-rehashing/>
3. Citation for question 5: <https://www.geeksforgeeks.org/hashing-set-3-open-addressing/#:~:text=Like%20separate%20chaining%2C%20open%20addressing,copying%20old%20data%20if%20needed>).