Quang Lam

1. Hash value is the integer value of an object the computer stores which can be used to uniquely identify data.

2. Hash value can be used in a hash table by dividing it to the length of the list and using the remainder as the index in the list.

3. After having the location, we check the value at that location to see if it equals to the item we're looking for. If not, we check the values of the locations following it until we reach the end of the chain.

4. Collision resolution is needed because as we divide a hash value (which is an integer) to the length of the list, the remainders are not always unique (always in the range from 0 to (length of the list - 1)). So without the resolution, we might have items having a same location and overwrite each other.

5. The set datatypes contains a group of unique values. A map datatype contains a set of unique keys that map to associated values.

6. We can create a dictionary with the values as the keys of the dictionary. The values associated with those keys are not important. In this case, I would just set the associated values to True if the HashSet has the keys.

For the membership method, I can check the associated value at the certain key. If it is True, return True. If it is not defined or is False, return False.

For the add method, I can just set the associated values at the certain key to True.

7. The load factor is the value of dividing the number of items stored in the list by its length. It allows us to decide when to extend the length of the list in relative to the number of items. And like Question 2 &

Question 4, as as we divide the hash value to the length of the list and use the remainder as the index in the list, the remainders are not always unique and always in the range from 0 to (length of the list - 1). The smaller the load factor, the smaller the number of items in relative to the length of the list and the range of the remainders we use a indexes is more sparse - which makes collision less likely to happen. And with less collisions, it's less likely for us to search through the list for check for membership.

8. Rehashing is to transfer the values of a list into a new list with different length. And the values need to be hashed again because the length of the length of the list changes.

9. Memoization is a technique that can be used when functions may get called more than once with same arguments and return the same values over and over again.

10. False. It's different from the Fibonacci function. To calculate the factorial of a certain number, it would call each of factorial functions of smaller numbers ONCE. For example, to calculate factorial(5), it would subsequently call factorial(4), factorial(3), factorial(2), factorial(1).