**Q1**. Explain the difference between a dictionary and hast table.

Dictionary is an abstract data structure which has an unordered collection of data as a set of key-value pairs.

Hash table is a data structure that implements a dictionary.

**Q2**. Explain what a hashing algorithm is used for in terms of hash tables.

The hashing algorithm uses the data to generate the hash value, which is then used to represent the key or position of the data in the hash table.

**Q3**. Explain what is meant by open addressing (collision resolution).

According to Wikipedia1:

“Open addressing (closed hashing) is a method of collision resolution in hash tables. With this method a hash collision is resolved by probing or searching through alternative locations in the array (the *probe sequence*) until either the target record is found, or an unused array slot is found, which indicates that there is no such key in the table.”

Types of probing2:

* Insert(k): Keep probing until an empty slot is found. Once an empty slot is found, insert k.
* Search(k): Keep probing until the slot’s key doesn’t become equal to k or an empty slot is reached.
* Delete(k): Delete operation is interesting. If we simply delete a key, then the search may fail. So, slots of deleted keys are marked specially as “deleted”.   
  The insert can insert an item in a deleted slot, but the search doesn’t stop at a deleted slot.

Probe sequences:

* Linear probing

The algorithm searches through the hash table from the original hash location until an unoccupied location is found.

If the original hash function is:

hash(key) = key % (table size)

The rehash function will be:

rehash1(key) = (hash(key) + n) % (table size)

then the next iteration (n = n +1) is:

rehash2(key) = (rehash1(key) + n) % (table size)

Where n is the number of iterations

* Quadratic probing/ **mid-square** method

In this method, starting from the original hash location, if the location is occupied then algorithm searches for the**n2 th** slot in the **nth** iteration.

If the original hash function is:

hash(key) = key % (table size)

The rehash function will be:

rehash1(key) = (hash(key) + n2) % (table size)

then the next iteration (n = n + 1) is:

rehash2(key) = (rehash1(key) + n2) % (table size)

* Double hashing

The increments for the probing sequence are computed by using another hash function. Using another hash function hash2(key) to look for the n \* hash2(key) slot in the **n th** iteration.

If the original hash function is:

hash(key) = key % (table size)

The rehash function will be:

rehash1(key) = (hash(key) + n \* hash2(key)) % (table size)

then the next iteration (n = n + 1) is:

rehash2(key) = (rehash1(key) + n \* hash2(key))) % (table size)

Source of information

1. <https://en.wikipedia.org/wiki/Open_addressing>

2. https://www.geeksforgeeks.org/open-addressing-collision-handling-technique-in-hashing/