5. Tables, hash tables, hash functions, collisions.

1. The table - abstract data type

In computer science, a table is a collection of data elements whose complex elements contain:

- **key**, which must be unique
- **value**, which can be complex itself.

The elements are identified by their unique keys.

The operations that can be performed on a queue are:

Insert: Adds elements to the table.
Delete: Removes elements to the table.
Search: Finds elements in the table.

Self-ordering table vs. Direct-address table:

Both self-ordering tables and direct-address tables are data structures that allow for efficient storage and retrieval of key-value pairs. However, they differ in their approach to achieving this efficiency.

A self-ordering table is a data structure that dynamically rearranges its elements to improve access time for frequently accessed elements. Specifically, it keeps track of the order in which elements are accessed and uses this information to move frequently accessed elements to the front of the table. The goal is to minimize the number of table lookups needed to find the desired element.

On the other hand, a direct-address table is a data structure that uses an array to map keys to their corresponding values directly. Each key is mapped to a specific index in the array, and the corresponding value is stored at that index. The advantage of this approach is that access time is constant (O(1)) because the key can be used to compute the index in the array directly. However, this approach requires that the key space be finite and small enough to fit in memory since the size of the array must be proportional to the size of the key space.

2. Hash table

A hash table is a data structure used to store and retrieve key-value pairs. It is also known as a hash map or a dictionary. In a hash table, a key is mapped to a specific index in an array using a hash function. This index is then used to store or retrieve the corresponding value associated with the key.

3. Hash function

A hash function is a mathematical function that takes an input (usually a string or a set of data) and returns a fixed-size output called a hash value or hash code. The hash value is typically used to index into a hash table or a hash-based data structure, allowing for efficient storage and retrieval of data.

4. Collision:

When a hash function maps two or more keys to the same index in a hash table, it results in a collision. This can happen due to the limited size of the array, the nature of the hash function used, or the distribution of the keys. Collisions can lead to a decrease in performance, as additional operations are required to handle them. Different collision resolution techniques can be used to handle collisions, as discussed above. There are different ways to handle collisions, two of which are collision resolution by chaining and collision resolution by open addressing.

Collision resolution by chaining:

In this technique, each index in the hash table contains a linked list of key-value pairs that hash to the same index. When a collision occurs, the new key-value pair is added to the linked list at that index. When retrieving a value, the hash function is used to find the index, and then the linked list is searched to find the corresponding key-value pair.

The advantage of this technique is that it is simple to implement and can handle any number of collisions.

Collision resolution by open addressing:

In this technique, when a collision occurs, the hash function finds the next available index in the hash table. When retrieving a value, the hash function is used to find the index, and then the table is probed until the corresponding key-value pair is found.

In many cases, collision resolution by chaining is faster and uses less memory than the open addressing technique. For example, if you have a huge value part in your elements, the collision resolution by chaining uses memory exactly as much as necessary to store each value, but the open addressing table always has some free spaces allocated. Also, open addressing takes more steps during the search, than collision resolution by chaining, but if you use open addressing with chaining, you can reduce the number of steps.

5. Exercises

1. Put 1367 and 984 into a hash table of size m=5 by using a hash function: h(k) = k mod m.

2. Create a hash table with the following data and hash function, h(k) = k mod 7: 50, 700, 76, 85, 92, 73, 101
Collision resolution is solved by chaining

3. Create a hash table with the following data and hash function, h(k) = k mod 9:

19	blue
27	red
51	green
13	blue
37	brown
31	pink
25	orange

Collision resolution is solved by: a. Open addressing; b. Open addressing with chaining

4., Create a hash table with the following data and hash function, h(k) = k mod 12, with all the 3 technique,
a., chaining,
b., open addressing,
c., open addressing by chaining.

key value
21 blue
43 red
55 gray
8 brown
71 green
14 blue
33 yellow
27 yellow