Date: Aug 13, 2025

Collision

- Collision occurs when a hash function maps two keys to the same location.
- Ways to resolve collision:
    1. Open addressing
    2. Chaining

Collision resolution using open addressing

- When a collision takes place, open addressing computes new positions using a probe sequence and the next record is stored in that position.
- So, all the values are stored in the hash table.

Probing

- Probing is a process of examining memory locations in a hash table.
- Types of probing:
    1. Linear probing
    2. Quadratic probing
    3. Double hashing
    4. Rehashing

Linear probing

If a value is already stored at memory location $h(k)$ then the following hash function is used: $h(k, i) = [h'(k) + i] \mod m$ where m is the size of the hash table $h'(k)$ maybe $k \mod m$ and i is the probe number.

Searching a value using linear probing

1. Compute $h(k)$.
2. Examine the hash table in the order of indices $h(k)$, $h(k) + 1 \mod m$, $h(k) + 2 \mod m$ ... until one of the following happens.
    1. $h(k) + j \mod m$ has a key k.

- In this case, the desired value has been found.

2. The index $h(k) + j \mod m$ is empty.
   - In this case, the desired value is not in the table.
3. We return to the starting position $h(k)$.
   - In this case, the table is full and the table is not in the table.

Drawback

- The algorithm results in clustering therefore there is more risk of collision.
- Search time increases with the size of cluster.
- When a new value has to be inserted into a preoccupied position, it is inserted at the end of the cluster, thereby increasing the cluster length. This is called primary clustering.

Quadratic probing

$$h(k, i) = h(k) + (C_1 \times i) + (C_2 \times i^2) \mod m$$

where C_1 and C_2 are non-zero constants.

Disadvantage

- Even though it is free from primary clustering, secondary clustering may occur, i.e. if there is a collision between two keys, then the same probe sequence will be followed for both, so the probability of multiple collision increases as table becomes full.
- A sequence of successive probes may explore a small fraction of the table then we will not be able to find an empty location in the table despite of the fact that the table is not full.

Double hashing

If there is no collision, we use h_1 but to resolve collision we use:

$$h'(k) = h_1(k) + (i \times h_2(k)) \mod m$$

where h_1 is the primary hash function and h_2 is the secondary hash function

Advantage

- Minimizes primary and secondary clustering.

Rehashing

- When the hash table becomes nearly full, number of collision increases, degrading the performance of insertion and search operations.
- Then create a new hash table with double the size and move all entries of the original hash table to the new hash table by computing new hash values and inserting into the new hash table.

Advantage

- Simple.

Disadvantage

- Costly.

Collision resolution by chaining

- Each location in a hash table stores a pointer to a linked list that contains all the key values that were hashed to that location.

Advantage

- It remains effective even when the number of key values to be stored is much higher than the locations in the hash table.
- Unlike quadratic probing, it does not degrade when the table is more than half full.
- Absolutely free from clustering.
- Provides an efficient mechanism to handle collision.

Disadvantage

- Inherent disadvantages of linked lists.
    - Extra overhead to store pointers to next nodes.
    - Traversing a linked list has poor cached performance, making the process cache ineffective.