

## Open Addressing

### Open Addressing

in open addressing,

- Unlike separate chaining, all the keys are stored inside the hash table.
- No key is stored outside the hash table.

### Techniques used for open addressing are-

- Linear Probing
- Quadratic Probing
- Double Hashing

### Linear Probing

In linear probing,

- When collision occurs, we linearly probe for the next bucket.
- We keep probing until an empty bucket is found.

Given an ordinary hash function  $h'$ :  $U \{0, 1...m-1\}$ , the method of linear probing uses the hash function.

$$h(k, i) = (h'(k) + i) \bmod m$$

Where ' $m$ ' is the size of hash table and  $h'(k) = k \bmod m$ . for  $i=0, 1....m-1$ .

### Advantage-

- It is easy to compute.

### **Disadvantage-**

- The main problem with linear probing is clustering.
- Many consecutive elements form groups.
- Then, it takes time to search an element or to find an empty bucket.

### **Quadratic Probing**

In quadratic probing,

- When collision occurs, we probe for  $i^2$ th bucket in  $i^{\text{th}}$  iteration.
- We keep probing until an empty bucket is found.

Quadratic Probing uses a hash function of the form

$$h(k,i) = (h'(k) + c_1i + c_2i^2) \bmod m$$

Where (as in linear probing)  $h'$  is an auxiliary hash function  $c_1$  and  $c_2 \neq 0$  are auxiliary constants and  $i=0, 1...m-1$ .

$$h'(k) = k \bmod m$$

Double Hashing:

Double Hashing is one of the best techniques available for open addressing because the permutations produced have many of the characteristics of randomly chosen permutations.

Double hashing uses a hash function of the form

$$h(k, i) = (h_1(k) + i h_2(k)) \bmod m$$

Where  $h_1$  and  $h_2$  are auxiliary hash functions and  $m$  is the size of the hash table.

$h_1(k) = k \bmod m$  or  $h_2(k) = k \bmod m'$ . Here  $m'$  is slightly less than  $m$  (say  $m-1$  or  $m-2$ ).

### **Rehashing**

**rehashing means hashing again.** Basically, when the load factor increases to more than its pre-defined value (default value of load factor is 0.75), the complexity increases. So to overcome this, the size of the array is increased (doubled) and all the values are hashed again and stored in the new double sized array to maintain a low load factor and low complexity.