Hashing Technique

useful for searching

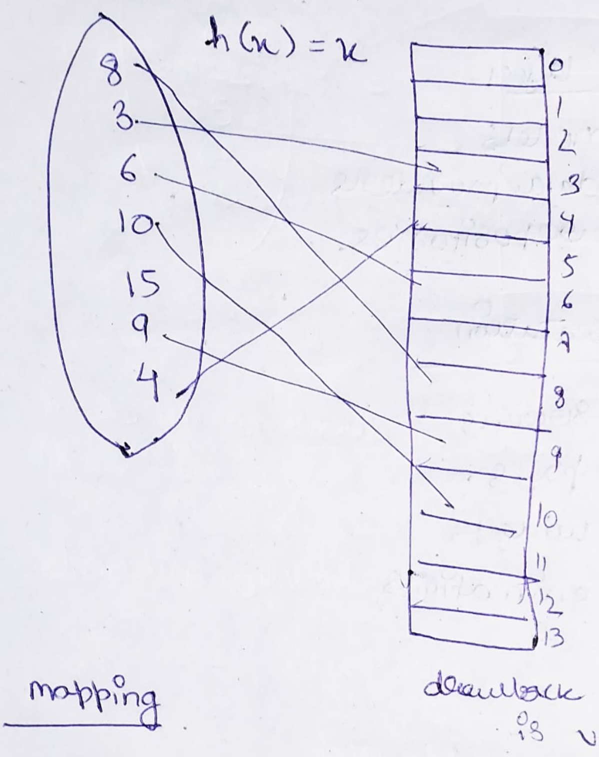
① linear – O(n)

② Binary – O(logn)

O(1)

keys : 8, 3, 6, 10, 15, 9, 4

key space



drawback is that space taken is very large.

mapping

functions

function

- one - one
- one - many
- many - one
- many - many

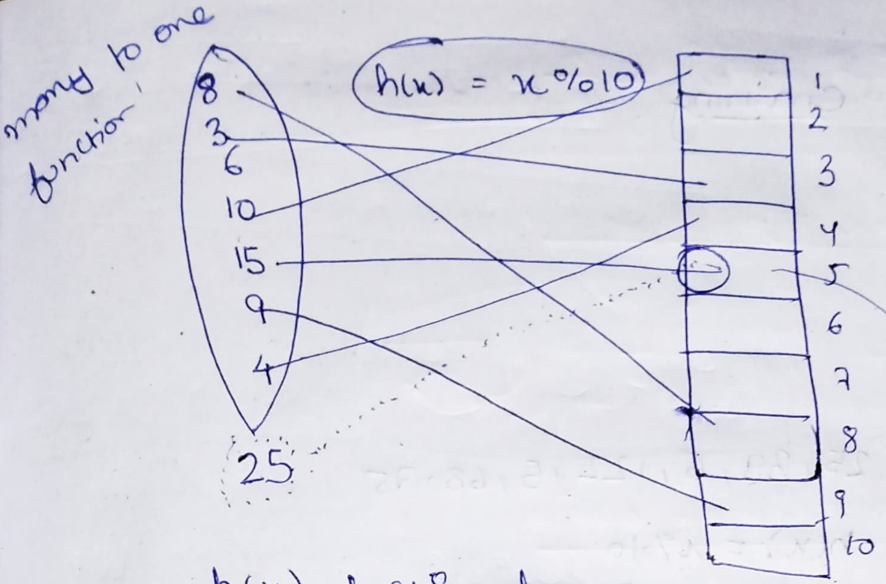
many to one function

this is

method

Open H

Closed



h(x) hashing function is responsible for space complexity of hash table.

Two keys are mapped in one position this is called collission.

method to avoid collision

Open Hashing → size can increase.

-> Chaining

Closed Hashing → fixed space.

Open addressing

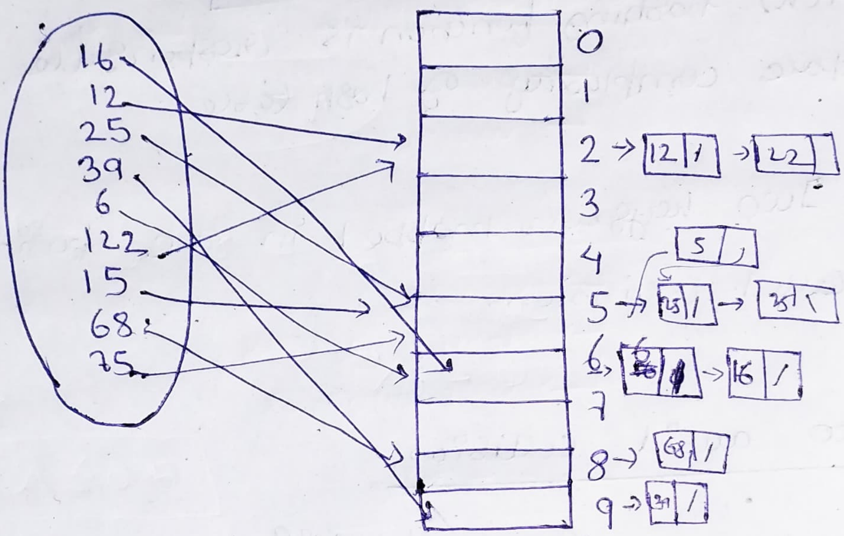
1. linear probing
2. Quadatric probing
3. Double Hashing

Chaining -

1. Insert
2. Search
3. Analysis
4. Delete.

keys : 16, 12, 25, 39, 6, 122, 15, 68, 75

$h(x) = x \% 10$



made up of array of chain or array of linked list.

Loading factor $\lambda = \frac{n}{size}$

Successful Search

Average successful time

$t = 1 + \lambda/2$

Average unsuccessful time

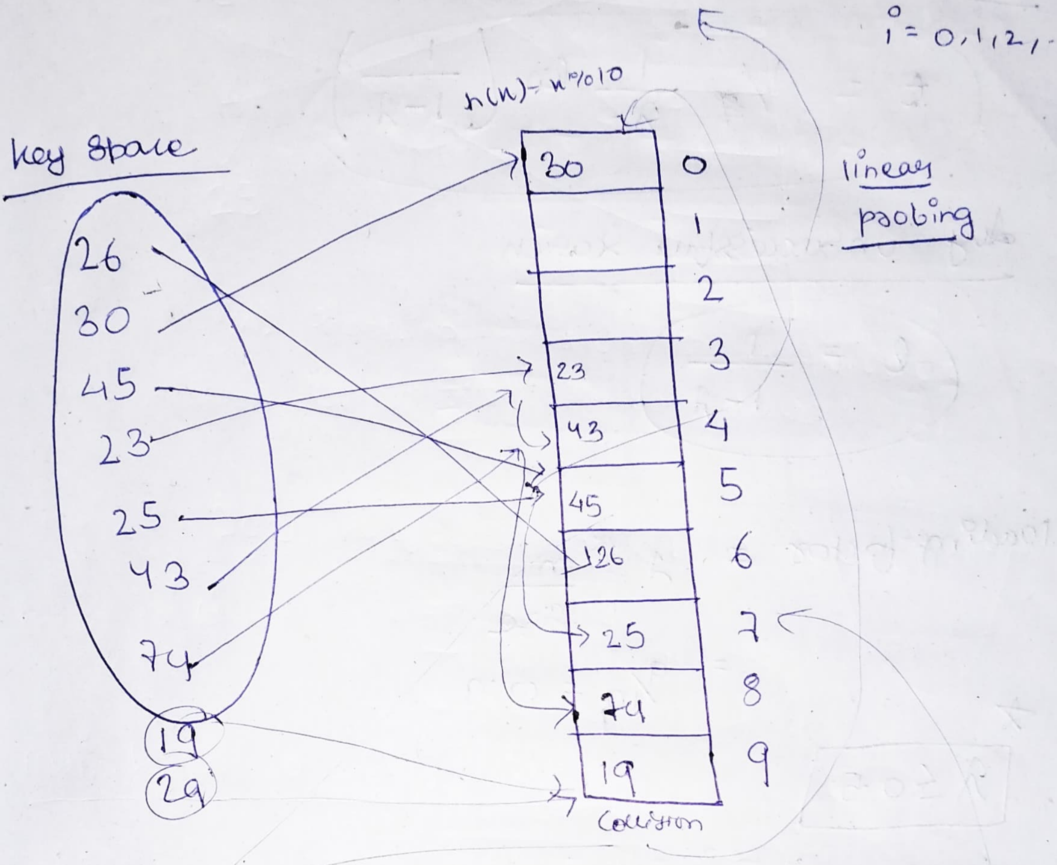
$t = 1 + \lambda$

Hashing Technique

(linear probing)

$h'(x) = (h(x) + f(i)) \% 10$ where $f(i) = i$

$i = 0, 1, 2, \ldots$



$h'(25) = (h(25) + f(0)) \% 10$
$(5 + 0) \% 10 = 5$ already occupied

$h'(25) = (h(25) + f(1)) \% 10$
$(5 + 1) \% 10 = 6$ already occupied

$h'(25) = (h(25) + f(2)) \% 10$
$(5 + 2) \% 10 = 7$ free

It will go cyclic

Search → take the hash function go to index then go on linearly until key is found or a empty space is found.

loading factor $\lambda = \frac{n}{size}$

Avg successful search

$$\boxed{t = \cancel{7} \frac{1}{\lambda} \ln\left(\frac{1}{1-\lambda}\right)}$$

Avg unsuccessful search

$$\boxed{t = \frac{1}{1-\lambda}}$$

loading factor $= \lambda = \frac{n}{size}$

$= 9/10 = 0.9$

☆ $\boxed{\lambda \leq 0.5}$

drawback

-> cluster formation
    -> A block of key at one place.

primary clustering

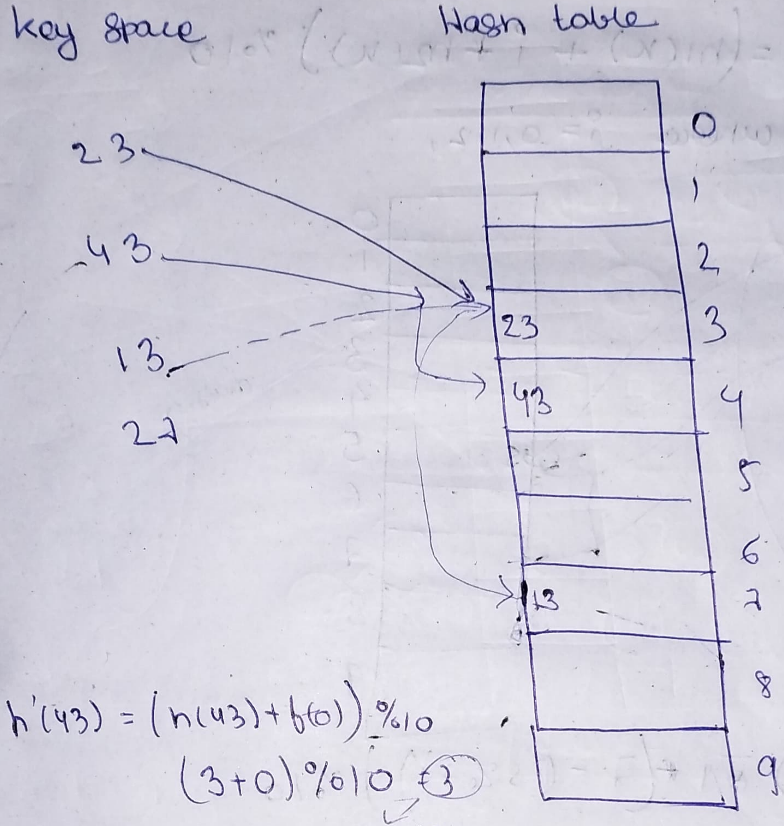
for deleting make a flag

Quadratic Probing

$h'(x) = (h(x) + f(i)) \% 10$ where

$f(i) = i^2$

$i = 0, 1, 2, \ldots$



$h'(43) = (h(43) + f(0)) \% 10$

$(3+0) \% 10 = \boxed{3}$

collision

$h'(43) = (h(43) + f(1)) \% 10$

$(3+1) \% 10 = \boxed{4}$

$h'(13) = h((43) + f(2)) \% 10$

$(3+4) \% 10 = 7$

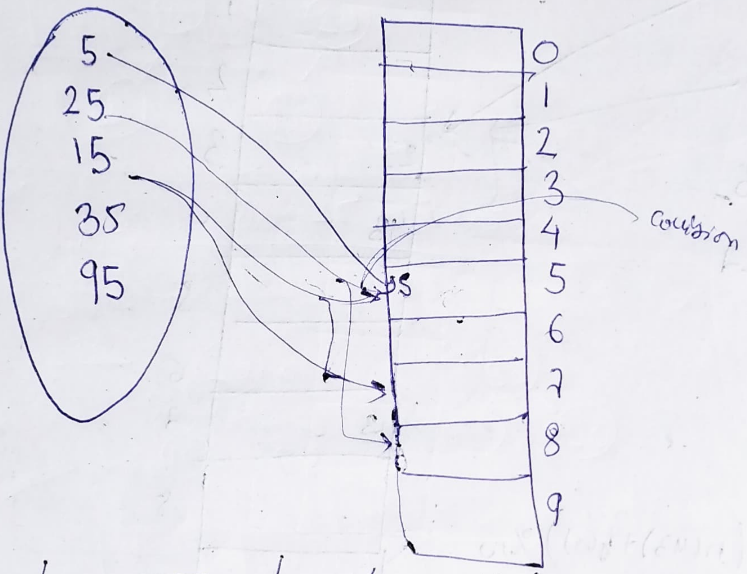
Hashing Technique (Double Hashing)

$h1(x) = x \% 10$

$h2(x) = 7 - (x \% 7)$

$h'(x) = (h1(x) + i * h2(x)) \% 10$

where $i = 0, 1, 2, \ldots$



$(h(25) + 1 * (7 - (25 \% 7))) \% 10$

$5 ~~+4~~ + 3 = 8$

~~$h'(15) = (5 + (2 * 6)) \% 10$~~

~~5+12~~