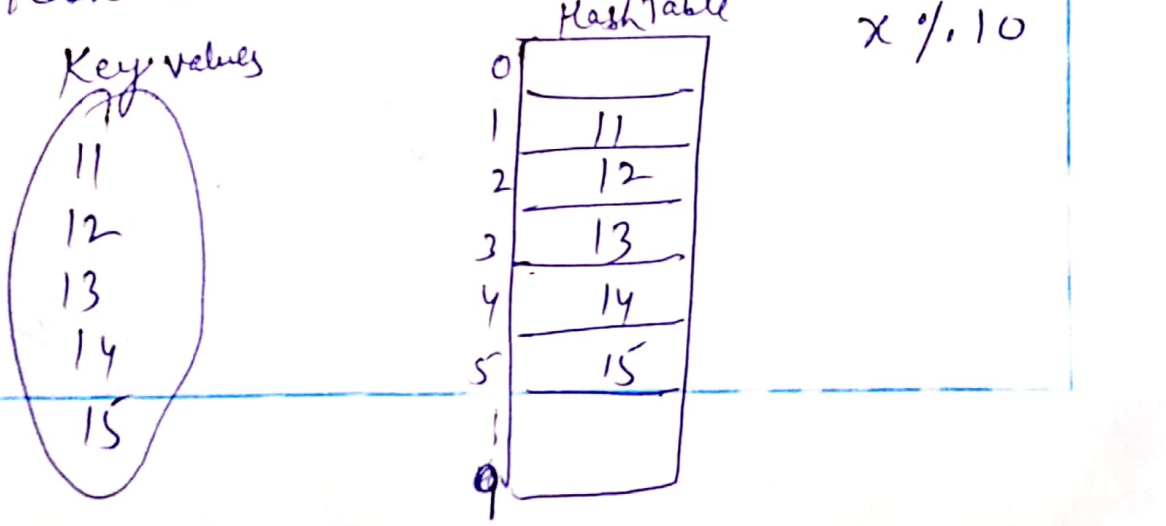
## Hashing :-

- Searching Technique based on Hash Table.
- It is the appln of a function to the Key values into the smaller range of relative address.
- Hashing is an important data structure which is designed to use a special function called the Hash function which is used to map a given value with a particular key for faster access of elements. The efficiency of mapping depends of the efficiency of the Hash function used.

Example:- Let a Hash function $H(x)$ maps the value $x$ at the index $x \% 10$ in an Array. If list of values is [11, 12, 13, 14, 15] it will be stored at position {1, 2, 3, 4, 5} in the array of Hash Table respectively.



NOTE!-

Linear Search ⇒ O(n)

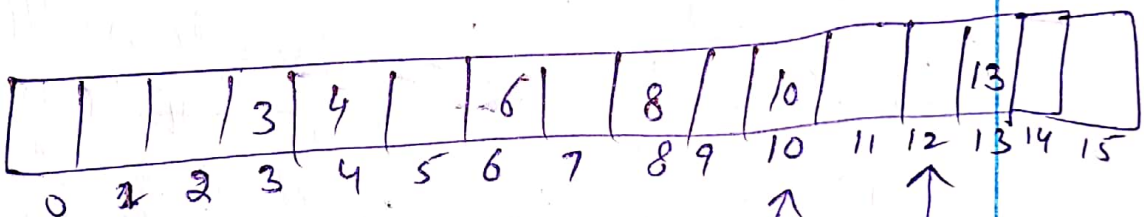
Binary Search ⇒ O(log n)
↳ (But list must be sorted first)

So, we want search method that take constant time. (i.e. O(1))

**Simple Hashing :-** we use here ideal Hash function.

$h(x) = x$

Keys ⇒ 8, 3, 13, 6, 4, 10



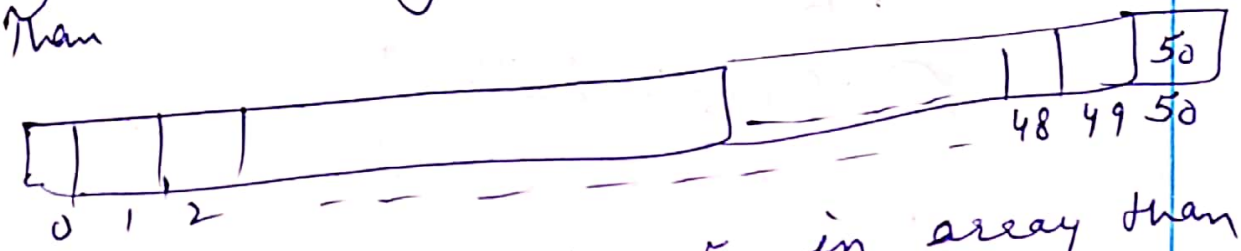

let Search Key ⇒ 10 — O(1)

let Search Key = 12 — O(1)

12 not found ∵ empty space

But problem in this, if we have an element (key value = 50)

Than




So, Here if we store 50 in array than we have 50 index in the array i.e very far away. So, lot of space is wasted in this method.

In this method, if you want to save time than lot of memory is wasted. Because we use large size array & most of the space are empty. So, wastage of memory is there.

So, we use different Hashfunction to overcome this type of problem.

---

Three methods are there to calculate the Hash function :-

1. Division Method
2. Folding method
3. Midsquare Method.

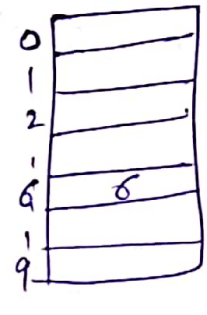
Let we use Here Divison Method :-

Example :- Key = 6, 26 $h(k) = k_i \% m$

m = 10
size of array / Hash Table

6 ⇒ [ 6 % 10 = 6 ]

26 [ 26 % 10 = 6 ]




Not Store here ∵ it is full

Called as Collision

To resolve these collision we have some techniques :-

Types of Hashing :-

① Open Hashing / Closed Addressing
one Method → Chaining

② closed Hashing / Open Addressing
3 methods
- Linear Probing
- Quadratic Probing
- Double Hashing Technique

# Open Hashing / Closed Addressing :-

One Method :-

## 1) Chaining Method :-

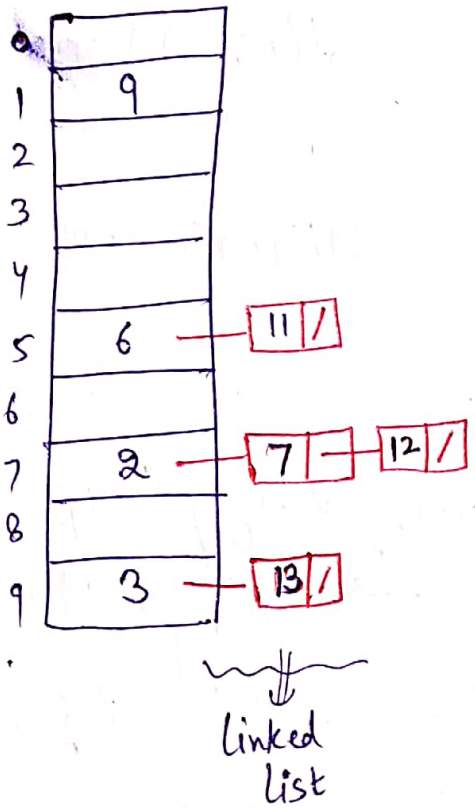
Q ⇒ A = 3, 2, 9, 6, 11, 13, 7, 12

$h(k) = 2k+3$ , $m = 10$ (size) , use division method & closed addressing to store these values.

If this fun.$^c$ is not given then directly use the key value. like 3 % 10 = 3

Division Method ⇒ $(k_i \% m)$

Sol$^M$ ⇒ Hash Table




| Key | location |
|---|---|
| 3 | [(2×3)+3] % 10 = 9 |
| 2 | [(2×2)+3] % 10 = 7 |
| 9 | [(2×9)+3] % 10 = 1 |
| 6 | [(2×6)+3] % 10 = 5 |
| 11 | [(2×11)+3] % 10 = 5 Collision |
| 13 | [(2×13)+3] % 10 = 9 |
| 7 | [(2×7)+3] % 10 = 7 |
| 12 | [(2×12)+3] % 10 = 7 |

CHAINING METHOD

Disadvantage :- ① clustering of elements

# LINEAR PROBING METHOD :-

Q⇒ A = 3, 2, 9, 6, 11, 13, 7, 12

$h(k) = 2k+3$, $m=10$, use division method & open addressing to store these values.

NOTE :- For open addressing ⇒ By default you will always use Linear Probing to resolve or minimize the collision.

Sol.ⁿ ⇒

| Key | location (u) | Probes |
|---|---|---|
| 3 | $[(2\times3)+3]\%10 = 9$ | 1 |
| 2 | $[(2\times2)+3]\%10 = 7$ | 1 |
| 9 | $[(2\times9)+3]\%10 = 1$ | 1 |
| 6 | $[(2\times6)+3]\%10 = 5$ | 1 |
| 11 | $[(2\times11)+3]\%10 = 5$ | 2 |
| 13 | $[(2\times13)+3]\%10 = 9$ | 2 |
| 7 | $[(2\times7)+3]\%10 = 7$ | 2 |
| 12 | $[(2\times12)+3]\%10 = 7$ | 6 |
| | | 16 Total Probes |

| | |
|---|---|
| Order of Element in Hash Table ⇒ | 13, 9, 12, –, –, 6, 11, 2, 7, 3 |

0 → 13 (④)
1 → 9 (⑤)
2 → 12 (⑥)
3
4
5 → 6
6 → 11
7 → 2 (①)
8 → 7 (②)
9 → 3 (③)

Search for key (12) free space

Probing ⇒ means "Searching", Here we search free space linearly.

Def.n :- Insert $k_i$ at the first free location from $(u+i)\%m$ where $i=0$ to $(m-1)$

Linear Probing In case of Collision

Key (11) ⇒ $(5+0)\%10 = 5$ ✗
⇒ $(5+1)\%10 = 6$ ✓

## Quadratic Probing :-

(Minimize the problem of clustering)

Insert $K_i$ at first free location from $(u+i^2)\ \%\ m$ , where $i=0$ to $(m-1)$

Q:) A = 3, 2, 9, 6, 11, 13, 7, 12

$h(k) = 2k+3$ , $m=10$ , use division method & quadratic probing to store these values

soln:)

| Key | location (u) | Probe |
|---|---|---|
| 3 | $[(2\times3)+3]\ \%10 = 9$ | 1 |
| 2 | $[(2\times2)+3]\ \%10 = 7$ | 1 |
| 9 | $[(2\times9)+3]\ \%10 = 1$ | 1 |
| 6 | $[(2\times6)+3]\ \%10 = 5$ | 1 |
| $K_i \rightarrow 11$ | $[(2\times11)+3]\ \%10 = (5)$ | 2 |
| 13 | $[(2\times13)+3]\ \%10 = (9)$ | 2 |
| 7 | $[(2\times13)+3]\ \%10 = (7)$ | 2 |
| 12 | $[(2\times12)+3]\ \%10 = (7)$ | 5 |

⇒ 15 total probes

| | |
|---|---|
| 0 | 13 |
| 1 | 9 |
| 2 | |
| 3 | 12 |
| 4 | |
| 5 | 6 |
| 6 | 11 |
| 7 | 2 |
| 8 | 7 |
| 9 | 3 |

Order :- 13, 9, –, 12, –, 6, 11, 2, 7, 3

for $K_i = 11$ ⇒

① $(u+i^2)\ \%\ m$
⇒ $(5+0^2)\ \%\ 10$
= 5 X

② $(5+1^2)\ \%\ 10$
= 6 ✓ free

Probes

for $K_i = 13$ :-

$(u+i^2)\ \%\ 10$
⇒ $(9+0^2)\ \%\ 10 = 9$ X
⇒ $(9+1^2)\ \%\ 10 = 0$ ✓

for $K_i = 7$ :-

$(7+0^2)\ \%\ 10 = 7$ X
$(7+1^2)\ \%\ 10 = 8$ ✓

for $K_i = 13$ :-

① ⇒ $(7+0^2)\ \%\ 10 = 7$ X
② ⇒ $(7+1^2)\ \%\ 10 = 8$ X
③ ⇒ $(7+2^2)\ \%\ 10 = 1$ X
④ ⇒ $(7+3^2)\ \%\ 10 = 6$ X
⑤ ⇒ $(7+4^2)\ \%\ 10 = 3$ ✓

Total ⑤ Probes

# Double Hashing Technique :-

Insert $k_i$ at first free place from $(u+v*i)\%m$ where $i=0$ to $(m-1)$

Q) $A = 3,2,9,6,11,13,7,12$

$h_1(k) = 2k+3$, $m=10$, use Division method & double hashing technique to insert these element where $h_2(k) = 3k+1$.

Soln ⇒

$v = h_2(k) \% m$

| Key | location (u) | v | Probes |
|---|---|---|---|
| 3 | $[(2\times3)+3]\%10 = 9$ | — No collision Comes | 1 |
| 2 | $[(2\times2)+3]\%10 = 7$ | — " | 1 |
| 9 | $[(2\times9)+3]\%10 = 1$ | " | 1 |
| 6 | $[(2\times6)+3]\%10 = 5$ | " | 1 |
| 11 | $[(2\times11)+3]\%10 = (5)$ | 4 (Collision occurs) | 3 |
| 13 | $[(2\times13)+3]\%10 = (9)$ | Not insert 13 in Hash Table ∵ $v=0$ | Reason → Next Page |
| 7 | $[(2\times7)+3]\%10 = (7)$ | 2 Not able to Insert '7' ∵ No free Space acc. to rule. | |
| 12 | $[(2\times12)+3]\%10 = (7)$ | 7 | 2 |

| | |
|---|---|
| 0 | |
| 1 | 9 |
| 2 | |
| 3 | 11 |
| 4 | 12 |
| 5 | 6 |
| 6 | |
| 7 | 2 |
| 8 | |
| 9 | 3 |

NOTE:- When no collision occur, simply put key value in Hash Table & when collision occur then we apply $h_2(k)$.

for $k_i = 11$ :-

$v = h_2(k) \% m$

$h_2(k) = 3k+1$

$\Rightarrow h_2(11) = 3\times11+1 = 33+1$

$h_2(11) = 34$

$v = 34 \% 10$

$v = 4$

Now $(u+v*i)\%10$

① $\Rightarrow (5+4*0)\%10$ = 5 ✗

② $\Rightarrow (5+4\times1)\%10 = 9$

③ $(5+4\times2)\%10$

$13\%10 = (3)$ ✓

Order of Element :-

—, 9, —, 11, 12, 6, —, 2, —, 3

Now, for $K_i = 13$ :- Collision Occurs

$h_2(K) = 3K+1$ , $v = h_2(K) \% m$

$h_2(13) = (3 \times 13 + 1)$ $\Rightarrow$ $v \Rightarrow 40 \% 10$

$= 40$ $\boxed{v = 0}$

* imp

Then, $(u + v * i) \% m$

$i=0 \Rightarrow (9 + 0 * 0) \% 10 = 9$ X

$i=1 \Rightarrow (9 + 0 * 1) \% 10 = 9$ X

$\boxed{0 * \text{any no.} = 0}$

therefore it always produce "9" again & again.

imp

So, we cannot insert '13' in this Hash Table, Although we have free spaces in Hash Tables but it is not necessary that all your Keys inserted in Hash Tables. Because sometime we are not able to find out the free location.