Hashing

Why we need Hashing

- Hashing can achieve O(1) time complexity
- Hash Table is created to stores elements in which hashing is performed
- Hash function

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

Size of Hash Table

key to be stored

address in hash table

Hash Function

if $m = 5$
$k = 20, 21, 22, 23, 32$

$h(20) = 20 \bmod 5 = 0$
$h(21) = 21 \bmod 5 = 1$
$h(22) = 22 \bmod 5 = 2$
$h(23) = 23 \bmod 5 = 3$
$h(32) = 32 \bmod 5 = 2$

collision occured

Hash Table

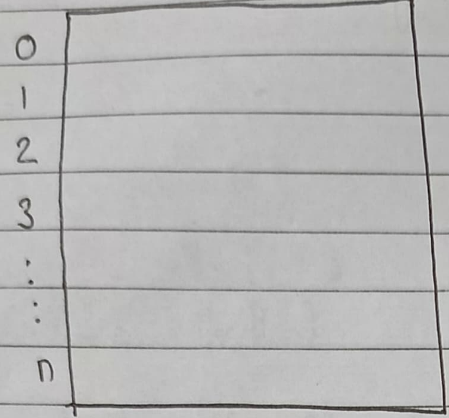
Hash Table Array

Index	element
0	20
1	21
2	22
3	23
4	~~30~~
	⋮

We cannot put 32 at address 2 as it is already occupied, ~~this~~ & two values of hash function are same ie value of 22 & 32 and this process is called collision.

Page No.
Date

Solution to Collission is DAT
Direct Address Table



- Diff b/w Hash Table & DAT

$\Rightarrow$ Hash Function - It is a mathematical formula used to ~~make the~~ map a given key within hash table

Types of Hash function

$\Rightarrow$ ~~Direct~~ Division Method - $h(k) = k \bmod m$

$\Rightarrow$ Mid square Method - k^2 ie $k=25 \Rightarrow k^2 = 625$ remove ~~elem~~ values from left & right and value at centre is the index of the element in this case K^1 ~~is~~ 25 and K^2 is 625 after removing ~~values~~ ~~from~~ the left & right most value which is 6 & 5, we got 2 as index of element. If $K^2 = 6024$ then index is 0 (Remove 1 from Left) & (Remove 2 from Right)

$\Rightarrow$ Folding Method - $K = 12, 34, 56$

$$\begin{array}{r} 12 \\ 34 \\ 56 \\ \hline 102 \end{array}$$

reject carry 1
and index is 02 or 2 for your element.

Callision

When more than one key point to the same slat in Hash table then it is called callission

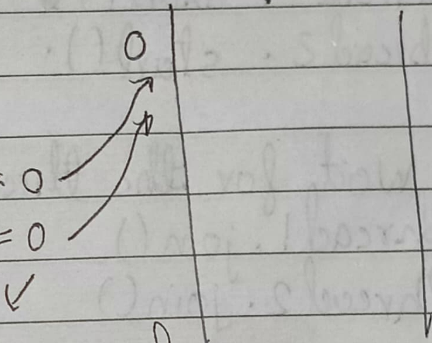
Hash Table

$h(k) = k \bmod m$

$k = 20, 30$

$h(20) = 20 \bmod 5 = 0$

$h(30) = 30 \bmod 5 = 0$

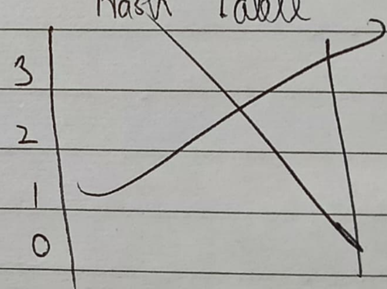


callission occured

Callision Resalution Technique
- Chaining Method
- Prabing Method

⇒ Chaining Method

Hash Table



$k = 20, 30, 40, 50, 11, 21, 31$

$h(k) = k \bmod m$

$h(20) = 0$

$h(30) = 0$

$h(40) = 0$

$h(50) = 0$

$h(11) = 1$

$h(21) = 1$

$h(31) = 1$

3	
2	
1	— [11] → [21] → [31] → NULL
0	~~20~~ — [20] → [30] → [40] → [50] → NULL

In Hash Table address is stored of the starting node. In this case address of 20 is stored at index 0 of Hash Table.

Probing
- Linear Probing
- Quadratic Probing
- Double Hashing

→ Linear Probing

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

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

m = No. of slots in Hash Table

i = Probe No. $= 0, 1, 2 \ldots$

$k = 20, 30, 32, 41, 51$

$h'(20) = ((20 \bmod 5) + 0) \bmod 5 = (0+0) \bmod 5 = 0$

$h'(30) = ((30 \bmod 5) + 0) \bmod 5 = (0+0) \bmod 5 = 0$

We cannot store 30 at index 0 as it is occupied by 20. So Now we are going to again call probe function for $h'(30)$ where $i = 1$

$h'(30) = ((30 \bmod 5) + 1) \bmod 5 = (0+1) \bmod 5 = 1$

There was collision b/w 20 & 30 key. And it is solved using prob function

~~Chaining Method~~

1	30
0	20

Problem of Linear Probing

- In probing in order to solve the problem of collision we calculate the index of hash table using Linear probing function. ~~We~~ While calculating the index we start by comparing the values from the start ie 0, 1, 2 Ex. This process is known as Primary Clustering

Quadratic Probing

$h'(k) = (h(k) + C_1 i + C_2 i^2) \bmod m$

C_1, C_2 = constants

$h(k) = k \bmod m$

$i = 0, 1, 2 \ldots$

$C_1 = 1, C_2 = 3$

keys = 20, 30, 32, 41, 51

1) $h'(20) = ((20 \bmod 5) + 1(0) + 3(0)^2) \bmod 5$
$= ((0) + 0 + 0) \bmod 5$
$= 0$

2) $h'(30) = ((30 \bmod 5) + 1(0) + 3(0)^2) \bmod 5$
$= ((0) + 0 + 0) \bmod 5$
$= 0$

Again run for $h'(30)$ but $i = 1$

3) $h'(30) = ((30 \bmod 5) + 1(1) + 3(1)^2) \bmod 5$
$= ((0) + 1 + 3) \bmod 5$
$= 4 \bmod 5$
$= 4$

4) $h'(32) = ((32 \bmod 5) + 1(0) + 3(0)^2) \bmod 5$
$= ((2) + 0 + 0) \bmod 5$
$= 2 \bmod 5$
$= 2$

⇒ Problem with Quadratic Probing
Secondary Clustering

Double Hashing

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

$h_1(k) = k \bmod m$

$h_2(k) = k \bmod m$

h_1 & h_2 are hash function we can use any Method like divide function, Mid square, Folding Method

$Q = h_1(k) \quad m = 5$

$h_2(k) \quad m = 3$

Q) Question on Double Hashing 5

given keys 71, 29, 38, 61, 100 map these keys in a hash table of size 5 with $h_1 = k \bmod$ & $h_2 = k \bmod 4$

~~$h(29) =$~~

$\Rightarrow$ $h(k) = [(h_1(k) + i\, h_2(k)]\ \text{mod}\ 5$

$h(71) = [(71 \bmod 5) + 0\,(71 \bmod 4)]\ \text{mod}\ 5$

$h(71) = [1 + 0]\ \text{mod}\ 5$

$h(71) = 1$

$\Rightarrow$ $h(29) = [(29 \bmod 5) + 0(29 \bmod 4)]\ \text{mod}\ 5$

$h(29) = 4 + 0(1)]\ \text{mod}\ 5$

$h(29) = 4$

$\Rightarrow$ $h(38) = [(38 \bmod 5) + 0(38 \bmod 4]]\ \text{mod}\ 5$

$h(38) = (3 + 0]\ \text{mod}\ 5$

$h_1(38) = 3$

$\cancel{h(38) = [(38 \bmod 5) + 1(38 \bmod 4)] \bmod 5}$

$\cancel{h(38) = 3}$

$\Rightarrow h(61) = [(61 \bmod 5) + 0(61 \bmod 4)] \bmod 5$

$= (1 + 0) \bmod 5$

$= 1$ callission

$\Rightarrow h(61) = [(61 \bmod 5) + 1(61 \bmod 4)] \bmod 5$

$= [1 + 1] \bmod 5$

$= 2$

$\Rightarrow h(100) = [(100 \bmod 5) + 0(100 \bmod 4)] \bmod 5$

$= 0 + 0 \bmod 5$

$= 0$

Hash Table using double hashing

4	29
3	38
2	61
1	71
0	100