Hashing

Problem: Suppose we're storing employee records,
primary key : telephone no.
Task : 1) Insert 2) Search and 3) Delete

Solution:

1) Use array : time - O(n) - search
   * if sorted : time - O(log n) - search
   insert, delete > costly

2) Use Linked list: search - O(n)

3) Balanced Binary Search Tree:
   insert, search, delete → O(log n)

4) Hash table: insert, search, delete → O(1)

* **Hash function:** maps a big number or string to a small integer that can be used as index in hash table.

Ex: $h(x) = x \bmod 7$

2 keys results same value → collision

$x = 8 \quad \therefore h(x) = 8 \% 7 = 1$

$x = 1 \quad \therefore h(x) = 18 \% 7 = 1$

** Characteristics of good hash function:

— efficiently computable

— equal distribution of keys

Collision handling:

1) Open hashing / closed Addressing
   a) separate chaining

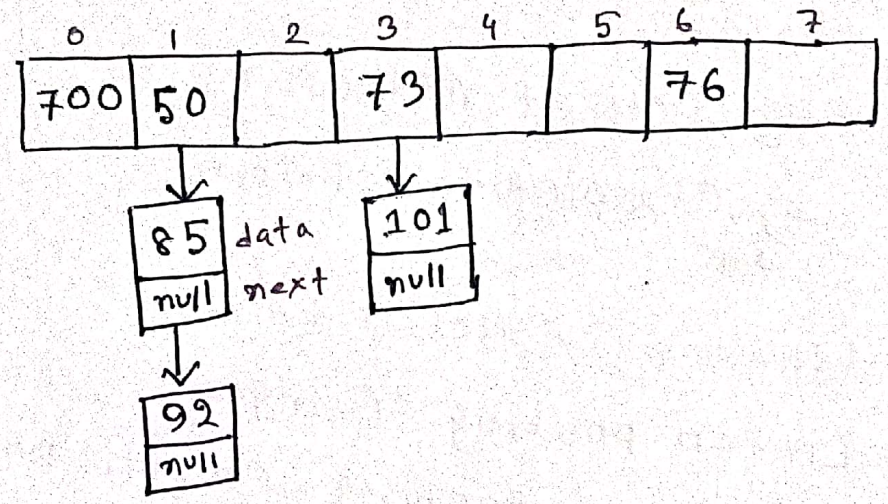
2) Open addressing / closed hashing
   a) Linear probing (single)
   b) Quadratic probing (multi-probe)
   c) Double Hashing

Separate chaining:

each cell of hash table
link to a linked list
- same hash value

Ex: hash function: $h(x) = x \% 7$ → index
keys: 50, 700, 76, 85, 92, 73, 101

Hash Table:



Step:
1) 50 % 7 = 1
2) 700 % 7 = 0
3) 76 % 7 = 6
4) 85 % 7 = 1 (linked list)
5) 92 % 7 = 1
6) 73 % 7 = 3
7) 101 % 7 = 3

Pros
- Simple implementation
- can add more elements to chain
- mostly used when the number of keys is unknown

Cons
- Memory waste
* Long chain - search O(n) - worst case
- Extra space for links,

Complexity (Time): $O\left(1+\frac{n}{m}\right)$

$n \rightarrow$ number of keys stored in table

Here, $n = 7$

m – number of slots in table

Here, $m = 8$

Linear Probing:

$h_i(x) = \{\text{Hash function}(x) + i\}\ \%$ table size

location

if the location is full $\rightarrow$ i++

probe: value of i+1

Ex: $h(k) = 2k+3$, size = 10

keys: 3, 2, 9, 6, 11, 13, 7, 12

	location	probe
1)	$\{2\cdot3+3+0\}\ \%\ 10 = 9$	1
2)	$\{2\cdot2+3+0\}\ \%\ 10 = 7$	1
3)	$\{2\cdot9+3+0\}\ \%\ 10 = 1$	1
4)	$\{2\cdot6+3+0\}\ \%\ 10 = 5$	1
5)	$\{2\cdot11+3+0\}\ \%\ 10 = 5$ $\{2\cdot11+3+1\}\ \%\ 10 = 6$	2
6)	$\{2\cdot13+3+0\}\ \%\ 10 = 9$ $\{2\cdot13+3+1\}\ \%\ 10 = 0$	2
7)	$\{2\cdot7+3+0\}\ \%\ 10 = 7$ $1 \quad = 8$	2

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

8) $\{2.12+3+0\}\%.10 = 7$ — Probe

1 = 8

2 = 9

3 = 0 → 6

4 = 1

5 = 2

Quadnadic probing:

$hi(x) = \{ \text{Hash function}(x) + i^v \}$ % table size

location

if location full → i++

probe : i+1

Ex: keys : 3, 2, 9, 6, 11, 13, 7, 12

table size : 10 ; $h(k) = 2k+3$

key	location	probe
3	$\{2.3+3+0^v\}\%.10 = 9$	1
2	$\{2.2+3+0^v\}\%.10 = 7$	1
9	$\{2.9+3+0^v\}\%.10 = 1$	1
6	$\{2.6+3+0^v\}\%.10 = 5$	1

Key	location	probe
11	$\{2.11+3+0^2\}\%10 = 5$	
	$\{2.11+3+1^2\}\%10 = 6$	2
13	$\{2.13+3+0^2\}\%10 = 9$	
	$\{2.13+3+1^2\}\%10 = 0$	2
7	$\{2.7+3+0^2\}\%10 = 7$	
	$\{2.7+3+1^2\}\%10 = 8$	2
12	$\{2.12+3+0^2\}\%10 = 7$	
	$\{2.12+3+1^2\}\%10 = 8$	
	$\{2.12+3+2^2\}\%10 = 1$	
	$\{2.12+3+3^2\}\%10 = 6$	
	$\{2.12+3+4^2\}\%10 = 3$	5

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

Double Hashing: use 2 hash functions

$$h_i(x) = \{\text{Hashfunc1}(x) + i * \text{hashfunc2}(x)\} \% \text{ table size}$$

$$\{0 \le i \le \text{tablesize} - 1\}$$

Ex: keys : 3, 2, 9, 6, 11, 13, 7, 12

size : 10

$h_1(k) = 2k + 3$

$h_2(k) = 3k + 1$

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

key	location	probe	probe
3	$\{2 \cdot 3 + 3 + 0(3 \cdot 3 + 1)\} \% 10 = 9$		1
2	$\{2 \cdot 2 + 3 + 0(3 \cdot 2 + 1)\} \% 10 = 7$		1
9	$\{2 \cdot 9 + 3 + 0(3 \cdot 9 + 1)\} \% 10 = 1$		1
6	$\{2 \cdot 6 + 3 + 0(3 \cdot 6 + 1)\} \% 10 = 5$		1
11	$\{2 \cdot 11 + 3 + 0(3 \cdot 11 + 1)\} \% 10 = 5$ $\{2 \cdot 11 + 3 + 1(3 \cdot 11 + 1)\} \% 10 = 9$ $\{2 \cdot 11 + 3 + 2(3 \cdot 11 + 1)\} \% 10 = 3$		3
13	$\{2 \cdot 13 + 3 + 0(3 \cdot 13 + 1)\} \% 10 = 9$ $\{2 \cdot 13 + 3 + 1(3 \cdot 13 + 1)\} \% 10 = 9$		—

7 $\{2 \cdot 7 + 3 + 0 (3 \cdot 7 + 1)\} \% 10 = 7$

$\{2 \cdot 7 + 3 + 1 (3 \cdot 7 + 1)\} \% 10 = 9$

$\{2 \cdot 7 + 3 + 2 (3 \cdot 7 + 1)\} \% 10 = 1$

$\{2 \cdot 7 + 3 + 3 (3 \cdot 7 + 1)\} \% 10 = 3$

4 = 5

5 = 7

6 = 9

7 = 1

8 = 3

9 = 5

12 $\{2 \cdot 12 + 3 + 0 (3 \cdot 12 + 1)\} \% 10 = 5$

$\{2 \cdot 12 + 3 + 1 (3 \cdot 12 + 1)\} \% 10 = 4$