3.4 - Hashing - 1

Sunday, November 23, 2025 4:47 PM

Searching

Linear → O(N)

Binary → O(log N)

O(1) → Hashing

→ Hashing is a technique which is used to perform searching in O(1).

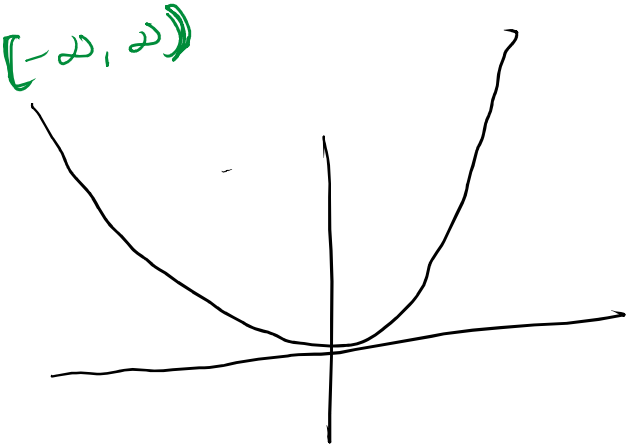
Basic Maths

$y = f(x)$ — independent variable; dependent variable

$y = x^2$

x	y
1	1
2	4
3	9

⇒ $y = x^2$ → parabola



⇒ $y = \sqrt{x}$ → $x >= 0$

[Domain] → valid inputs

$[0, \infty)$

→ output

[Range] → output

$y = x^2 \Rightarrow [0, \infty)$

#

$y = x \% E$

Domain → $(-\infty, \infty)$

Range → $[0, E-1]$

$x \% 4 \rightarrow \{0, 1, 2, 3\}$

$y = x \% 4$

x	$x \% 4$
0	0
1	1
2	2
3	3
4	0
5	1
6	2

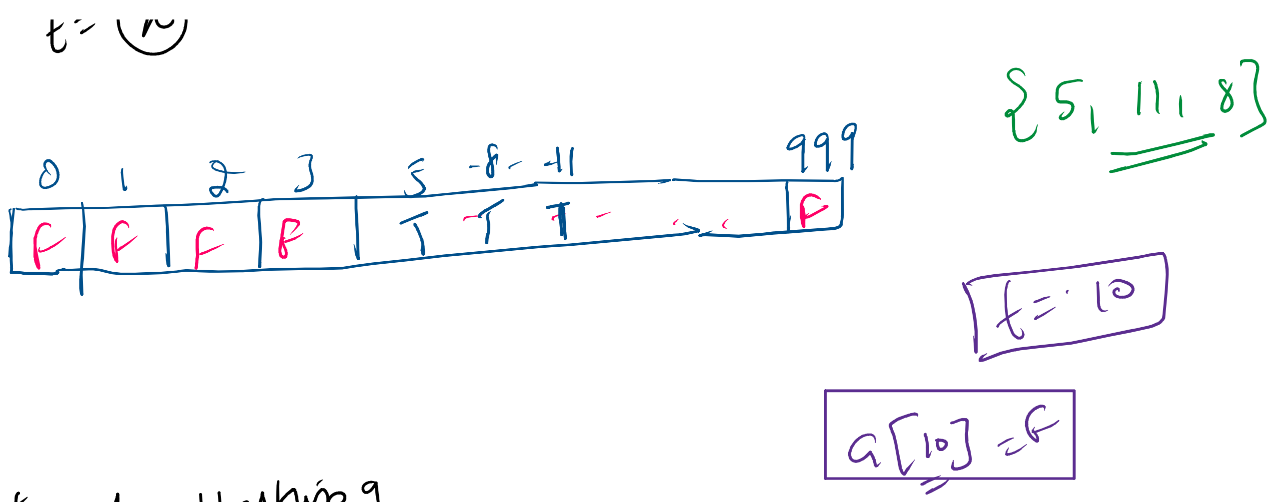
$[0, E-1]$

#

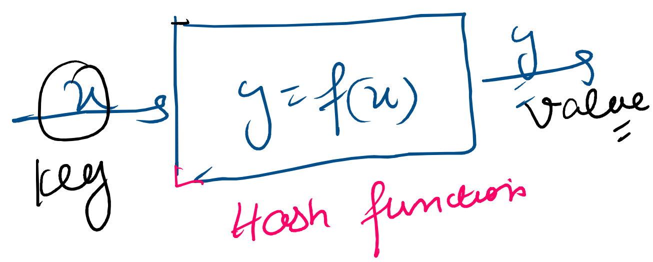
Array ⇒ [_ _ _ _ _] → O(N)

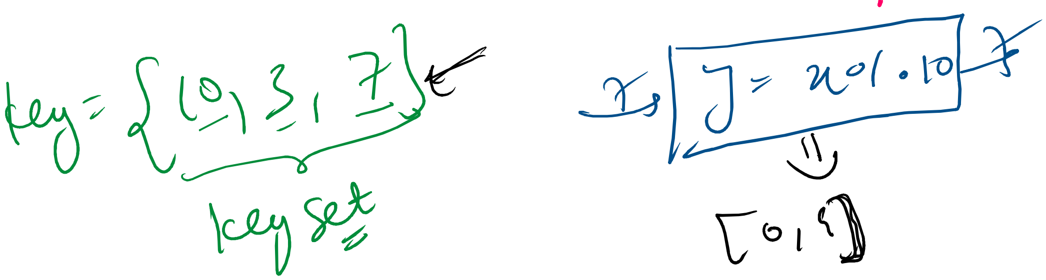
t = (x)

5, 11, 87



Concept of Hashing





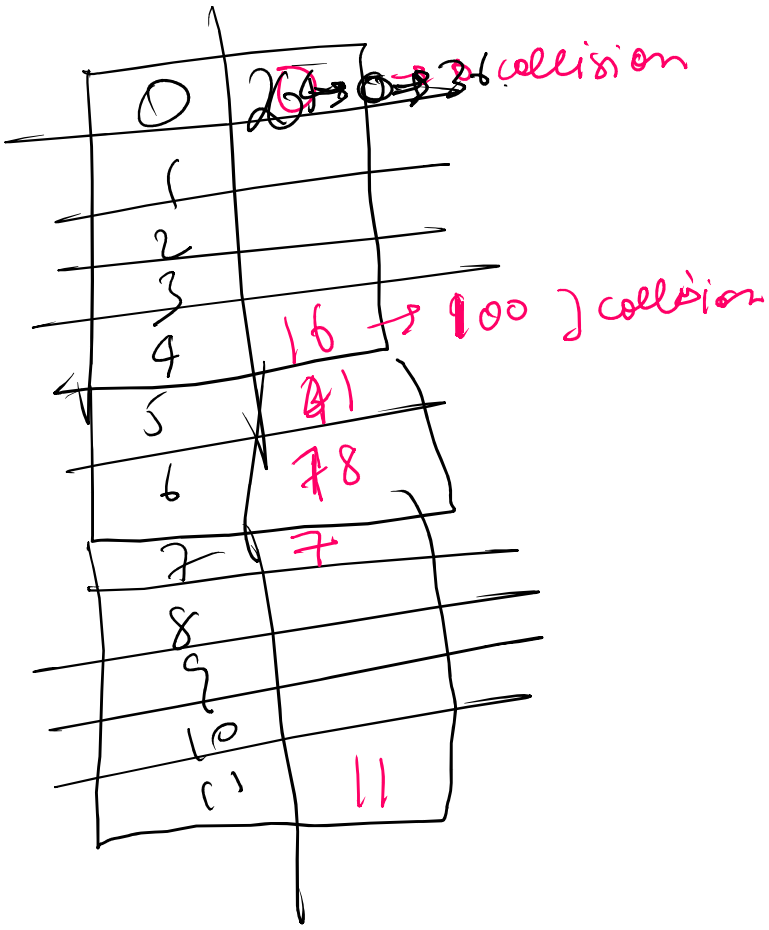
0	10
1	
2	
3	3
4	
5	
6	
7	7
8	
9	

- Size of Hash Table depends on Hash Function.
- If range of hash function is big, space will be wasted
- If range is very small, then collisions will increase.

Q) $h(x) = x \% 12$

$K = \{7, 16, 24, 78, 41, 100, 0, 11\}$

Design Hash Table, and count the no of collisions:



Load Factor

$$LoF = \frac{\text{No of entries in Hash Table}}{\text{size of the hash table}}$$

A good hash func^n should have load

A good hash func^{n} should have load factor ≤ 0.75

(A) If load factor goes beyond 0.75, we perform re-hashing.

How to resolve/handle collisions:

- open hashing
  - → Linked Hashing
- closed hashing
  - → Linear Probing
  - → Quadratic Probing

⇒ Linked Hashing

searching will become O(N).

⇒ Linear probing

⇒ Linear probing

$h(x) = x \% 10$

$h'(x) = [h(x) + i]$ $\quad i = 0, 1, 2 \cdots$

$h(x) = x \% 5$

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

$k = \{13, 14, 27, 5, 10, 20\}$

0	10
1	20
2	30
3	13
4	14
5	5
6	
7	27
8	
9	

→ Linear probing leads to clustering.

quadratic

$h'(x) = h(x) + i^2$