

grading

(Lorraine)

DS & Algo

1-10

Coding

1-10

Problem Solving

1-10

Communication

1-10

SMART

Find Min in Rotated Sorted Array

$l = 0$

$h = \text{len}(\text{arr}) - 1$

~~base case~~
if $\text{arr}[l] \leq \text{arr}[h]$
return $\text{arr}[l]$

while($l \leq h$):

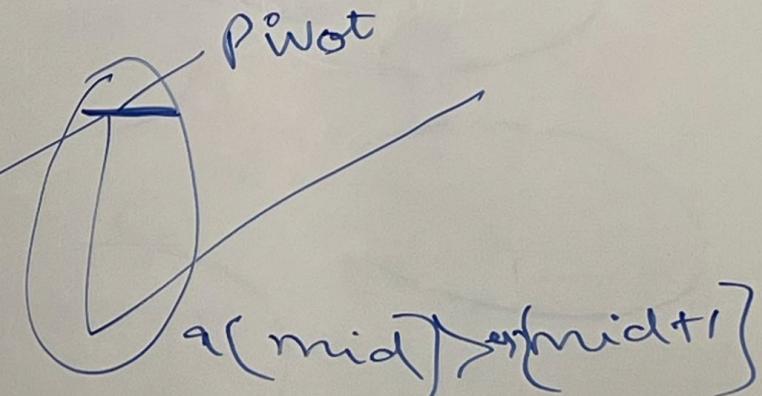
$$\text{mid} = l + (h-l)/2$$

if $\text{arr}[\text{mid}] > \text{arr}[\text{mid}+1]$
===== return $\text{arr}[\text{mid}+1]$

if $\text{arr}[l] \leq \text{arr}[\text{mid}]$ (left is sorted)
 $l = \text{mid}+1$

else $h = \text{mid}-1$

Pattern



$[17, 23, 35, 80, 91, 98, 0, 5, 13] [3, 4, 5, 1, 2]$

~~case 2~~

$[11, 13, 15, 17]$

$[4, 5, 6, 7, 0, 1, 2]$

$[8, 0, 2, 3, 4]$

SMART

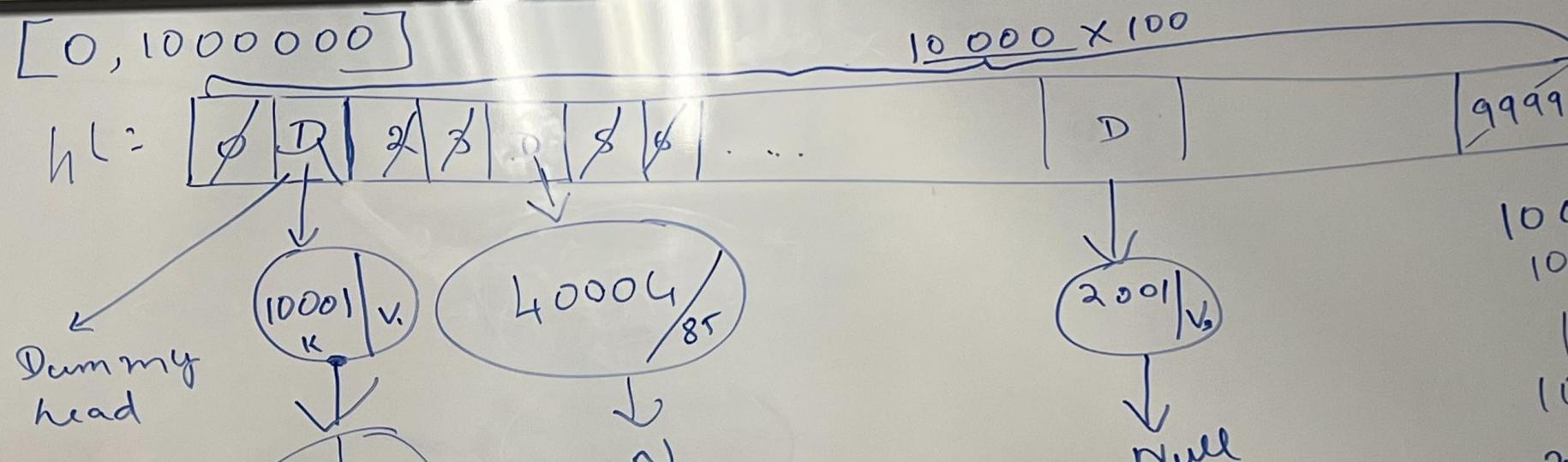
Hashmap

- $O(1)$ search
- Key is used as index, we use array
- Key \rightarrow index \rightarrow Hashing
- Collision
- CHT
- Linear Chaining
- Double Hashing (σ) $\propto \frac{1}{n^2}$

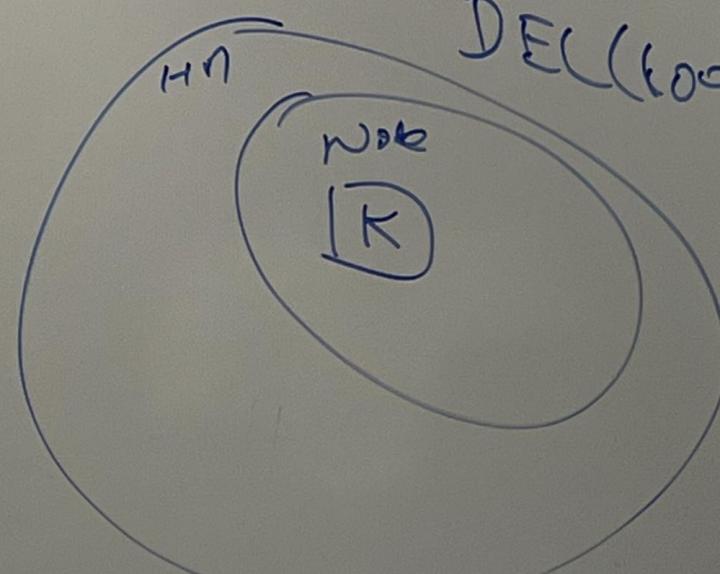
P. next = Self. Node

 (K_{10004}, V_{40004})

P. next. Val = Value



10001%
10000
1%
10,000
2001%
10000
40000%
10000
70001%
10000
DEL(40001)
60004%
10000
DEL(60004)

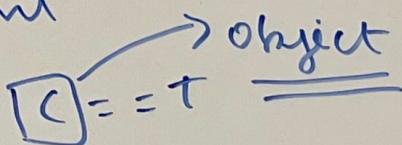


- Search in 2D matrix

- Find Min in Rotated Sorted -

- Find Peak element

t = 40001

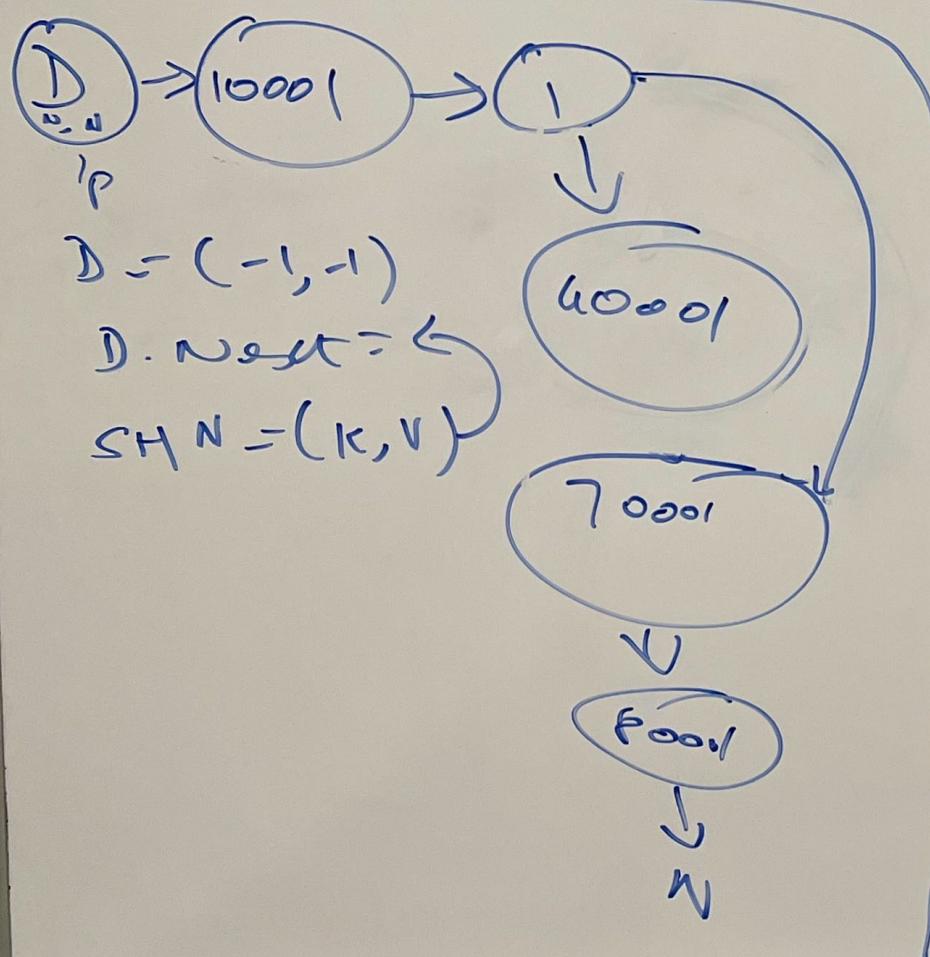


C.key == t : P.NSL.N

Scri - C.next.Val == t: C.N == C.N.N

(== C.next
find_node

Constructor
overloading



Class Node {

Node() {

default
constructor

$$Hf(x) = ((a^p \times R_1 \% x) \overset{P_2}{\times} R_2 \%)$$

Random Size of Array

$$40001 // 10000 \\ = 4$$

Class Node:

Node(int K, int V,

Node Next)

this.K = K

↳ Parameterized
constructor

this.V = V

this.Next = Next

def __init__(self, K, V):

self.Key = Key

self.Val = Val

self.Next = None

Node N1 = new Node(1, 80, N40001)
N2 = Node()

- Search in 2D matrix

- Find Min in Rotated Sorted -

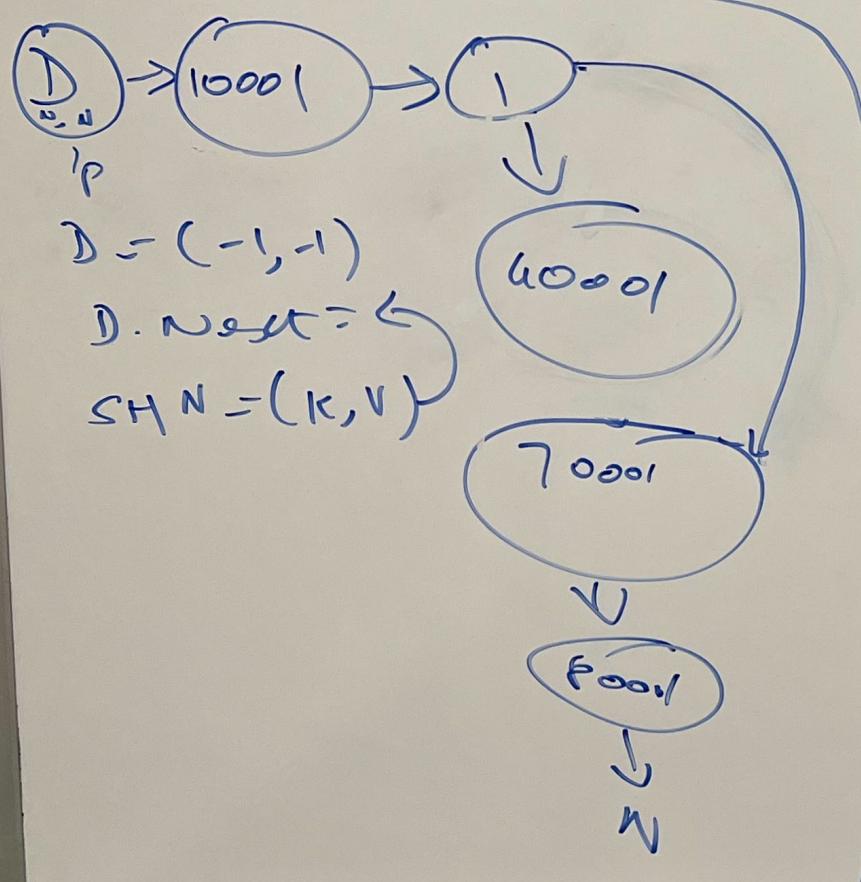
- Find Peak element

t = 40001

C == t
Object

C.key == t : P.N = C.N

Scri - C.next.Val == t : C.N = C.N.N
(== C.next
find_node
Constructor overloading



Class Node {

$$Hf(x) = \left(\left(\frac{P_1}{R} \times R \right) \% x \right)^{\frac{P_2}{b \times R_2 \%}} \\ \text{Random} \quad \text{Size of Array}$$

$$40001 // 10000 \\ = 4$$

Node() {

default
constructor

}

Node(int K, int V,

Node Next),

{

this.K = K

↳ Parameterized
constructor

this.V = V

this.Next = Next

}

Class Node:

def __init__(self, K, V):

self.Key = Key

self.Val = Val

self.Next = None

Node N1 = new Node(1, 80, N40001)
N2 = Node()