

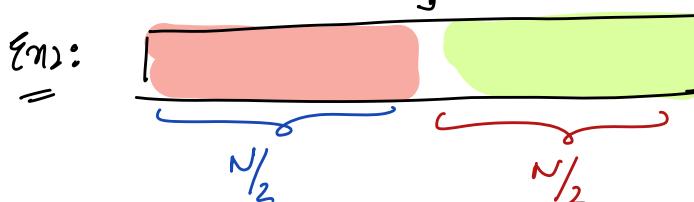
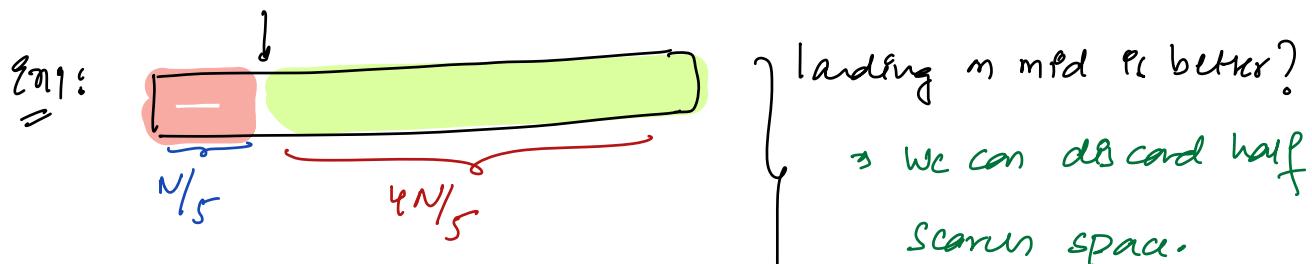
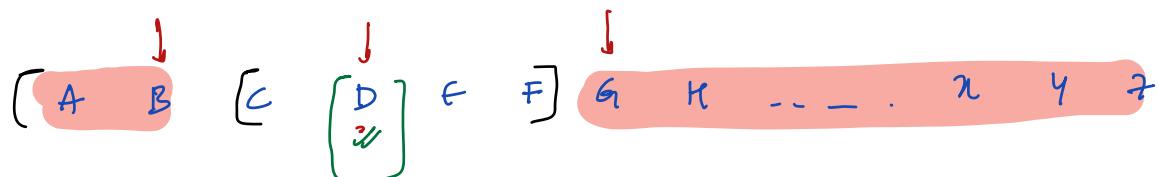
Search : ↗ (Target) : What to Search
 ↙ (Search Space) : Where to Search

word : Dict | News paper in book

content : Phone | Phone diary

→
 ↳ Arrangement in your search space which
 our searching easy.

Ex: Dict → {DOG}



Binary Search : Divide your search space into 2 halves &
Discard 1 half of space using some
conditions?

Given a sorted array with distinct elements, Search for index of an element k , if k not present return -1

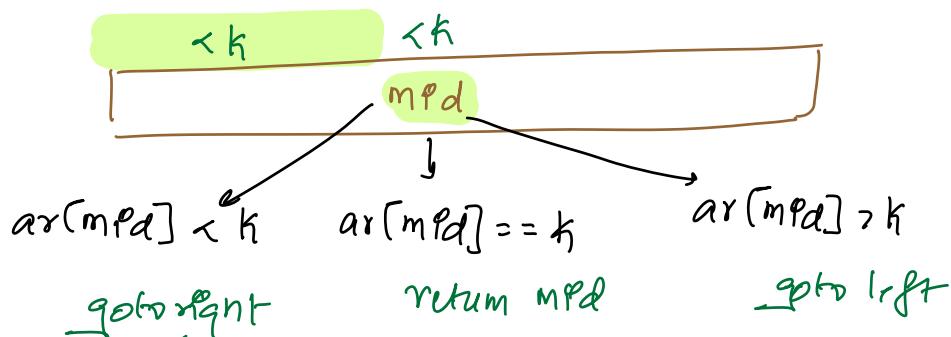
	0	1	2	3	4	5	6	7	8	9
$ar[10]$:	3	6	9	12	14	19	20	23	25	27

Sol:

linear search

TC: $\Theta(N)$

SC: $\Theta(1)$



0	1	2	3	4	5	6	7	8	9	}
3	6	9	12	14	19	20	23	25	27	$k = 12$

$$\underline{l} \quad \underline{h} \quad \underline{mpd} \rightarrow (l+h)/2$$

$0 \quad 9 \quad 4$ $\text{if } [ar[4] > 12] \{ h = mpd - 1 \}$ {decoding right}

$0 \quad 3 \quad 1$ $\text{if } [ar[1] < 12] \{ l = mpd + 1 \}$ {decoding left}

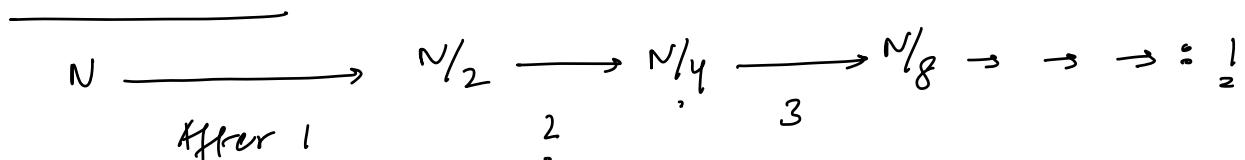
$2 \quad 3 \quad 2$ $\text{if } [ar[2] < 12] \{ l = mpd + 1 \}$ {decoding left}

$3 \quad 3 \quad 3$ $\text{if } [ar[3] == 12] \{ \text{return mpd} \}$

```

Pnt    searchEnd( ar[], N, h) {
    l = 0, h = n-1
    m - mid value
    while ( l <= h )
        m = (l + h)/2
        if ( ar[m] == h ) { return m }
        if ( ar[m] > h ) { // goto left
            h = m-1
        } else { // goto right
            l = m+1
        }
    }
    return -1;
}

```



$\Rightarrow \underline{TC} : (\underline{\log N})$

$\underline{SC} : \underline{\Theta(1)}$

Q) Given a sorted array, find floor of a given number k
 greater ele $i=k$ in $\text{arr}[]$

0 1 2 3 4 5 6 7 8 \leq

$\text{arr}[9] = \boxed{-5 \ 2 \ 3 \ 6 \ 9 \ 10 \ 11 \ 14 \ 18}$

$k=5 : 3$

$k=4 : 3$

$k=10 : 10$

$k=20 : 18$

$k=-7 : -5$

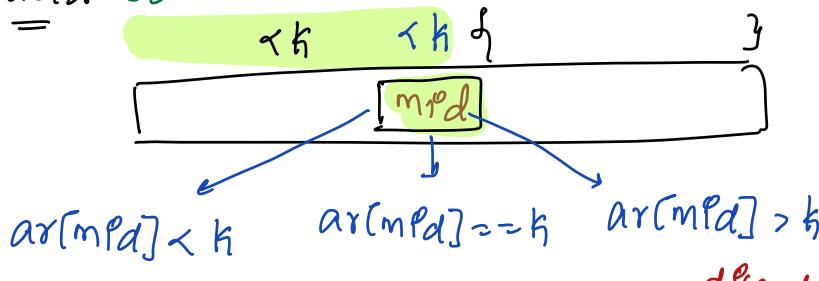
$-5 \neq -7 *$

return INT_MIN

Idea: $k=12$, } TC: $O(N)$ SC: $O(1)$

ans = $\text{INT_MIN}, -5, 2, 3, 6, 9, 10, 11$

Idea2: BS.



$\text{ar}[\text{mid}] < k$

$\text{ar}[\text{mid}] == k$

$\text{ar}[\text{mid}] > k$

$\text{ans} = \text{ar}[\text{mid}]$

return k

discard right
Search left

Search right

$0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8$ $k = 5$
 $ar[7] = \boxed{-5 \quad 2 \quad 3 \quad 6 \quad 9 \quad 10 \quad 11 \quad 14 \quad 18}$ $ans = INT_MIN$

$\frac{l}{0} \quad \frac{h}{8} \quad \frac{m}{4} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{if } (ar[4] > 5) \text{ right, } h = m - 1$
 descand

$0 \quad 3 \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{if } (ar[1] < 5) \text{ left}$ $ans = 2$
 $l = m + 1$

$2 \quad 3 \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{if } (ar[2] < 5) \text{ left}$ $ans = 3$
 $l = m + 1$

$3 \quad 3 \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{if } (ar[3] > 5) \text{ right}$ $h = m - 1$

$3 \downarrow \quad 2 \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{Break}$ $ans = 3$

$ans = INT_MIN$

$l = 0, h = n - 1$

while ($l <= h$)

$m = (l + h)/2$

$\text{if } (ar[m] == k) \text{ return } k$

$\text{if } (ar[m] > k) \{ // \text{goto left}$

$h = m - 1$

$\text{else} // \text{goto right}$

$l = m + 1, ans = ar[m]$

return ans ;

38) Given a sorted array of N Elements, find the frequency of a given element: k

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
-5	-5	-3	0	0	1	1	5	5	5	5	5	5	5	8	10	10	15	15

Sol1:

Iteratively get freq

TC: $\Theta(N)$ SC: $\Theta(1)$

Sol2:

Use hashmap & freq

TC: $\Theta(N)$ SC: $\Theta(N)$

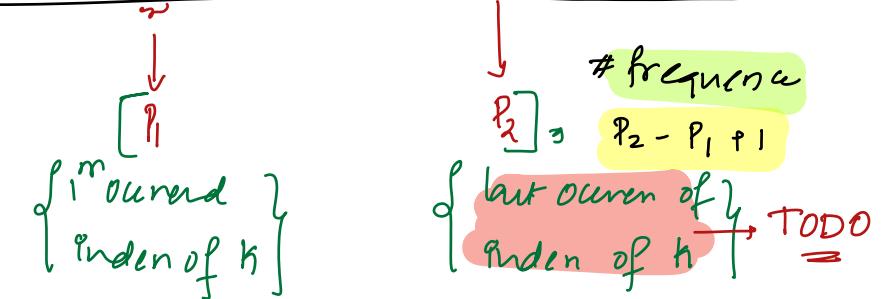
Sol3:

1) Use BS find end

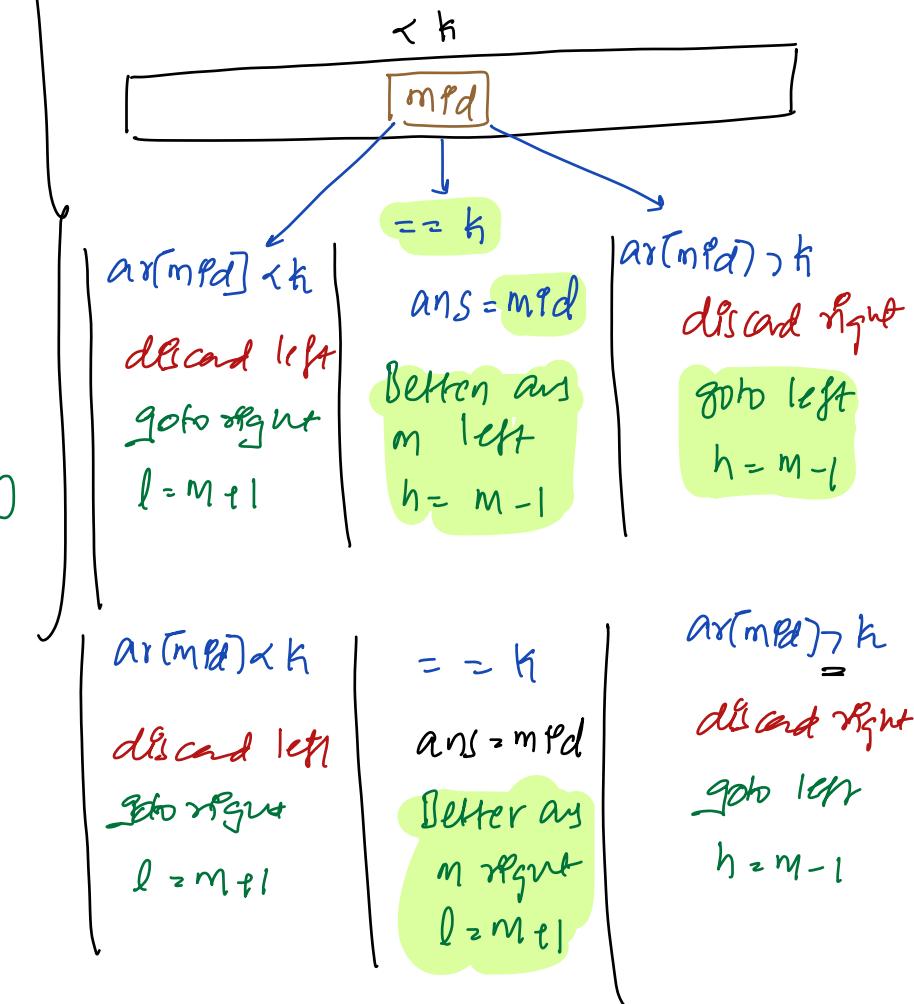
2) Use 2 phr get \rightarrow count

TC: $\Theta(\log N + N)$

SC: $\Theta(1)$



idcc for $p_1 \rightarrow 1^{\text{st}} \text{ occur index of } k$



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
-5	-5	-3	0	0	1	1	5	5	5	5	5	5	5	8	10	10	15	15

$$\underline{k=5}$$

<u>l</u>	<u>h</u>	<u>m</u>	<u>$ans = -1$</u>
0	18	9	if ($arr[9] == k$) $ans = 9$, go to right
0	8	4	if ($arr[4] < k$) go to right
5	8	6	if ($arr[6] < k$) go to right
7	8	7	if ($arr[7] == k$) $ans = 7$, go to left

$\leftarrow c \quad \text{Break} \right\rbrace \Rightarrow \underline{ans = 7} \rightarrow \text{in index of } 5$

$$TC \Rightarrow (\log N + \log N)$$

$$\begin{matrix} \swarrow \\ P_1 \end{matrix} \qquad \begin{matrix} \searrow \\ P_2 \end{matrix}$$

$$SC \Rightarrow \underline{\mathcal{O}(D)}$$

Q8) Given N distinct elements, Return any 1 peak element

An Element is said to be peak, If greater than the

Input $N \geq 2$

$ar[] :$	0 1 2 3 4 5 6	$\left\{ \begin{array}{l} ar[i-1] < ar[i] > ar[i+1] \\ ar[0] > ar[1] \\ ar[N-2] < ar[N-1] \end{array} \right.$
	-1 < 6 > 2 5 7 4 8 = = = = = = x ✓ x x ✓ x ✓	

\rightarrow we can return any
element from {6, 7, 8}

Idea1:

Iterate in array ar
for every element, check
if this peak or not

TC: $O(N)$ SC: $O(1)$

0 1 2 3 4 5 6
-1 6 2 5 7 4 8

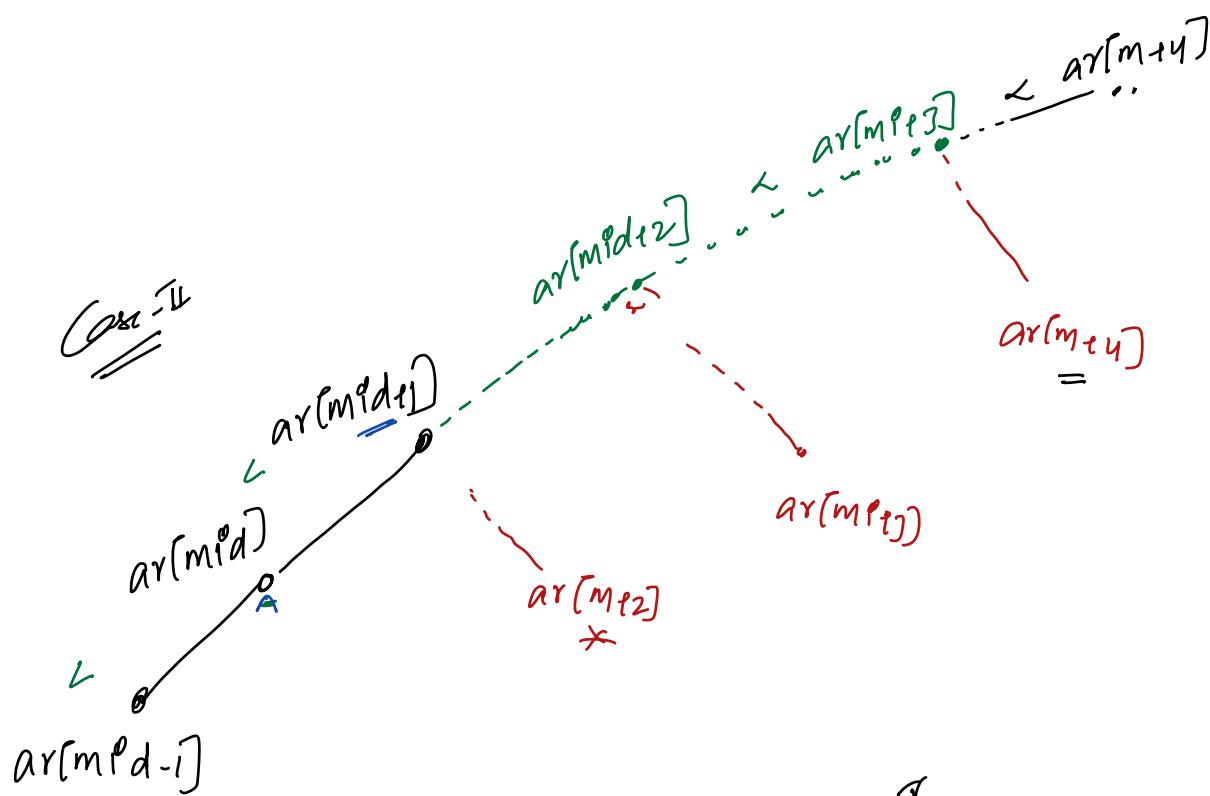
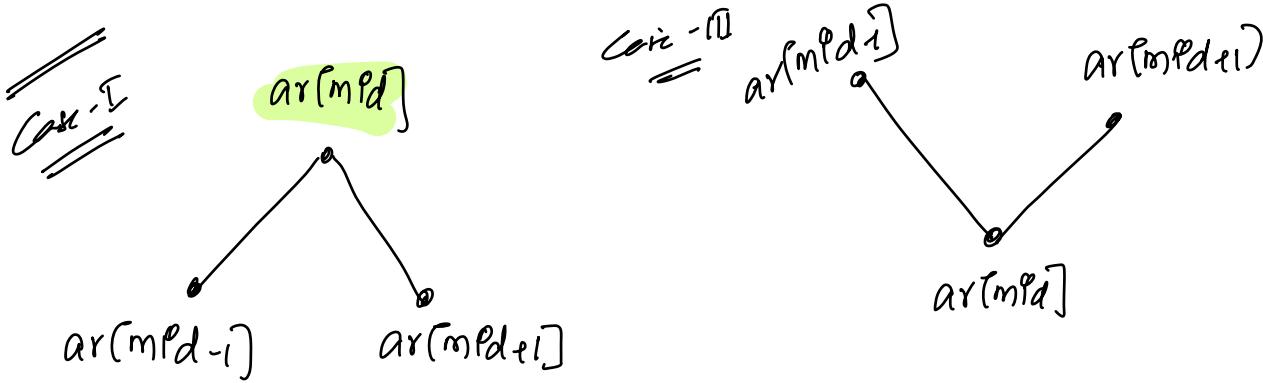
Idea3: $l = 0$ $h = 6$ $m = 3$ Is Peak
 x

$\//$ go both sides: $O(N)$

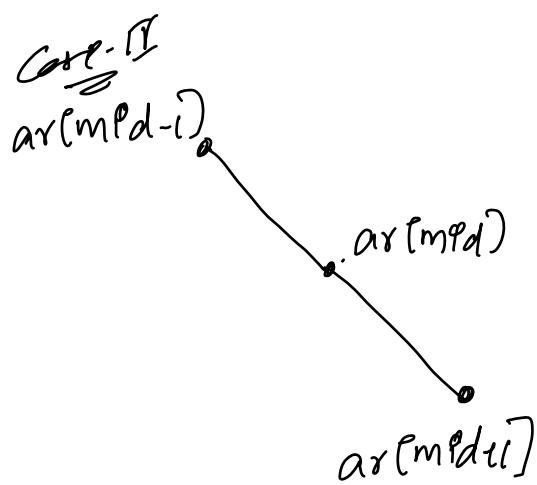
Idea2:

find max & return

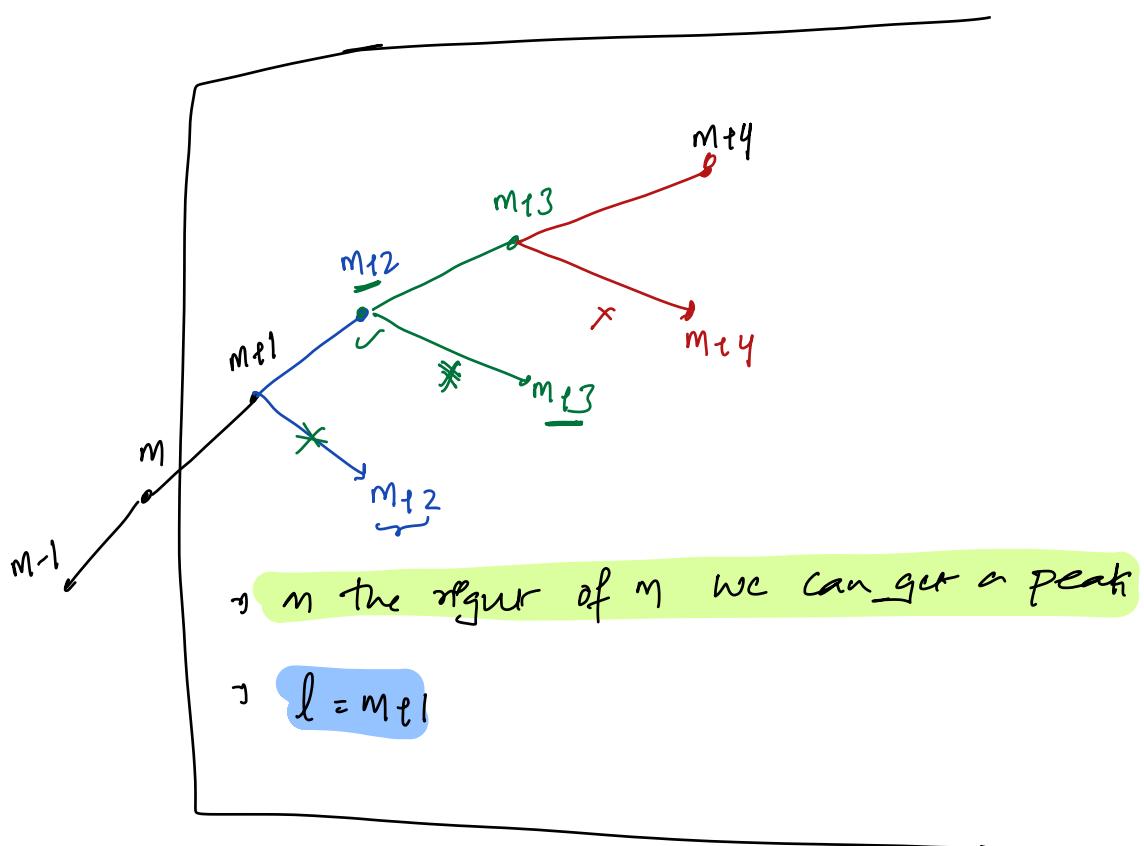
TC: $O(N)$ SC: $O(1)$



D



1



obs: When ever side is greater than mid in that side, atleast 1 peak is 100% present.

Pseudo Code

edge cases

```

    {
        if (ar[0] > ar[1]) return ar[0]
        if (ar[N-1] > ar[N-2]) return ar[N-1]
    }

```

$l = 0, 1; h = n - m - 2;$

while ($l \leq h$) {

$$m = (l + h)/2$$

if ($ar[mid-1] < ar[mpd]$ && $ar[mpd] > ar[mpd+1]$) {

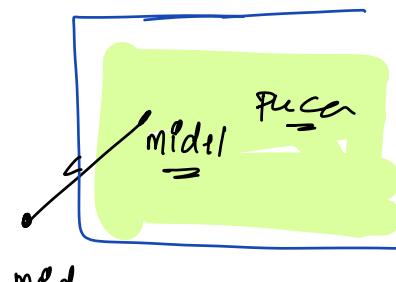
return $ar[mpd]$

if ($\underbrace{ar[mpd]}_0 < \underbrace{ar[mpd+1]}_0$)

$$l = m + 1$$

else {

$$h = m - 1$$



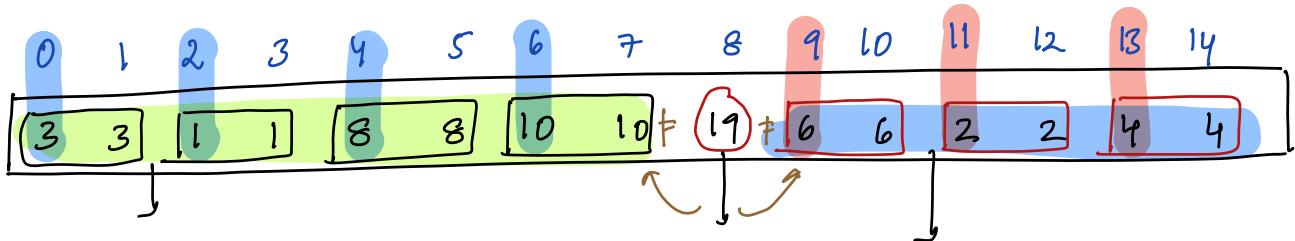
TC: $O(\log N)$
SC: $O(1)$

58

{ TODO } → { Thursday Session 3 }

Every element occurs twice except for 1, find unique element

Note: Duplicates are adjacent to each other.



If we land left:

goto right

All 1st occurrence they are
in even index

If we land in right

goto left

All 1st occurrence they are
in odd index

Doubts :