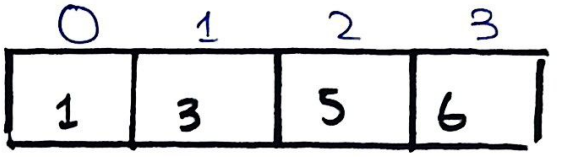
Q: 2

35. Search Insert Position

Ex : 01



left = 0;

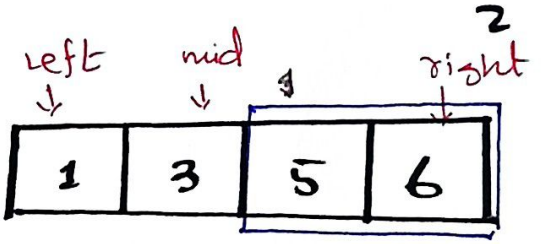
right = n-1; ∵ Here n=4

1st - iteration :

left = 0

r = n-1

mid = (L+r)/2 => (0+3)/2 => 1



if (num[mid] == t) return mid;

else if(nums[mid] < t) left = mid+1; *

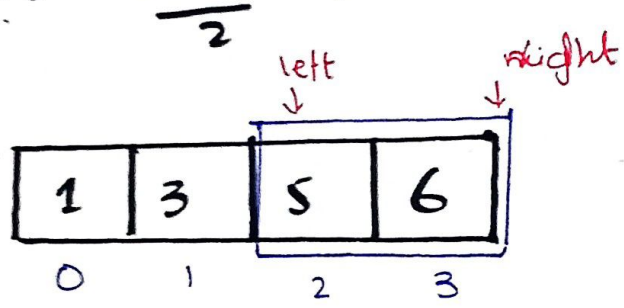
else (nums[mid] > t) right = mid-1;

Second - iteration:

left = 2

right = n-1 ⇒ 3

mid = $\frac{(2+3)}{2}$ ⇒ 2



if (nums[mid] == t) return mid;

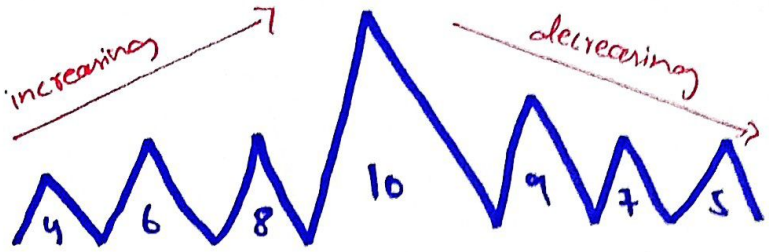
"

"

Q:2

852. Peak Index in a Mountain Array

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



left = 0

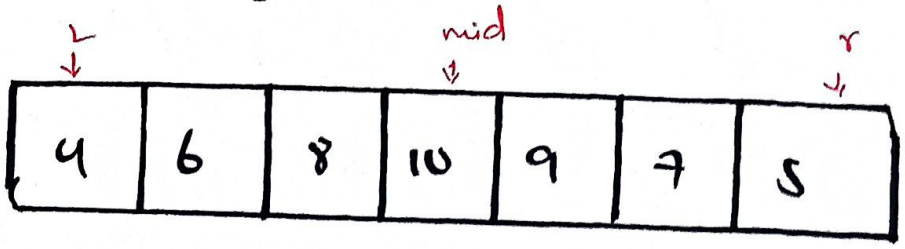
Right = n-1 => 6

1st - iteration :

L = 0

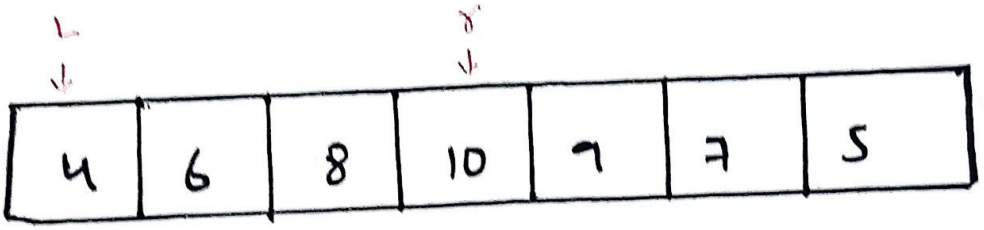
r = 6

mid = $\frac{(0+6)}{2}$ = 3



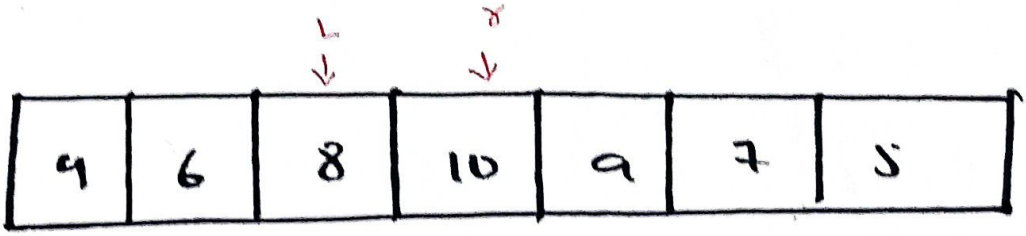
```
if( arr[mid] < arr[mid+1] )
        left = mid+1;
else if ( arr[mid] > arr[mid+1] ){
        right = mid;
```

r = mid;



2nd - iteration;

L = 0
r = 3
mid = 1



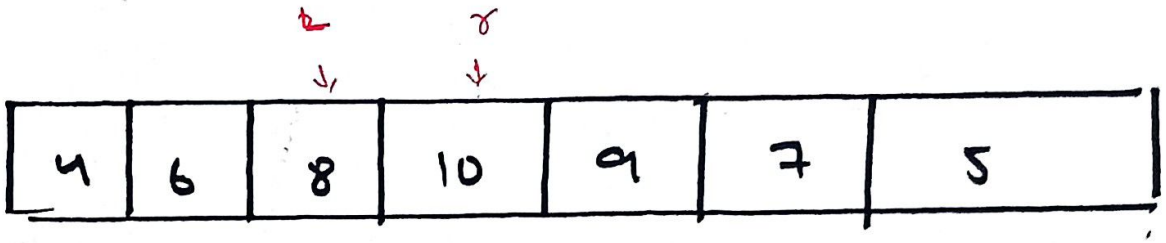
3-iteration:

~~l=0~~

l=2

r=3

m=2




```
if ( arr[mid] < arr[mid+1])
    left = mid+1
else if ( arr[mid] > arr[mid+1])
    right = mid;
```

4-iteration:-

~~l=3~~

~~r=3~~

~~m=3~~