

⑩ → Search insert position - #search

⑪ Question -

Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You must write an algorithm with $O(\log n)$ complexity.

⑫ Examples -

nums = [1, 3, 5, 6], target = 5

→ 2

nums = [1, 3, 5, 6], target = 2

→ 1

nums = [1, 3, 5, 6], target = 7

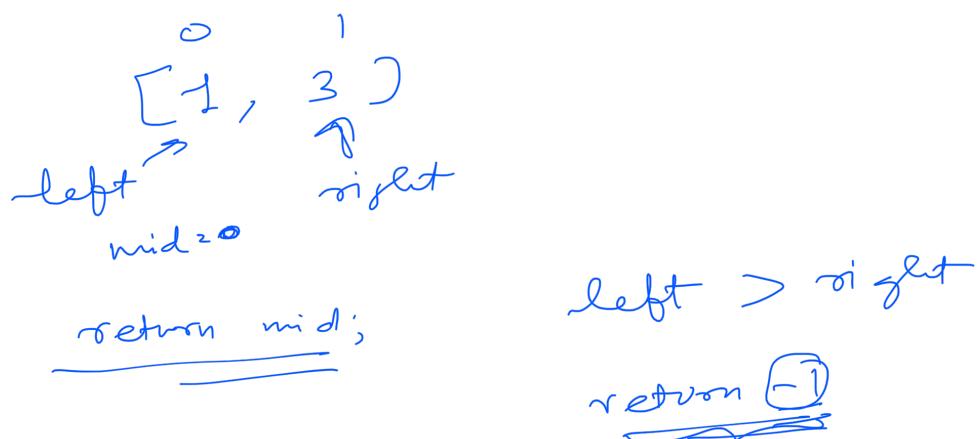
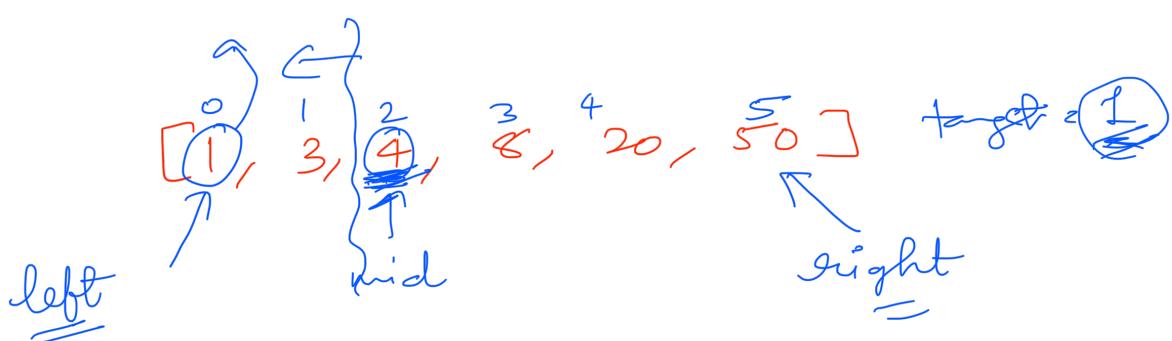
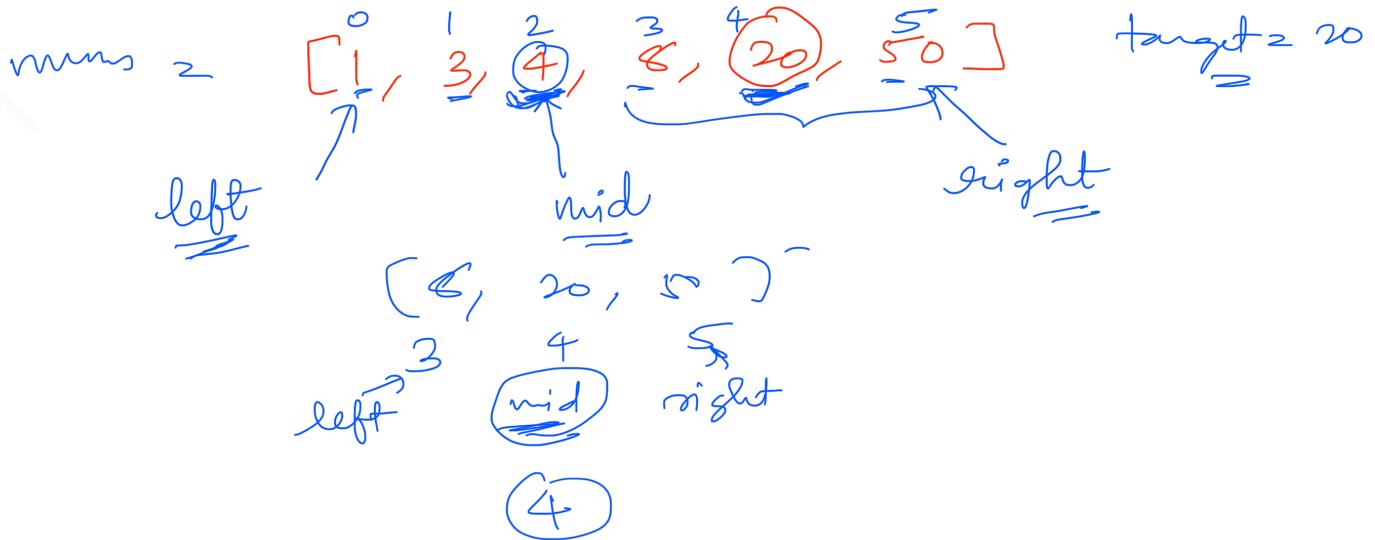
→ 4

⑬ Intuition -

$O(\log n)$ | Search position | Sorted array dis.

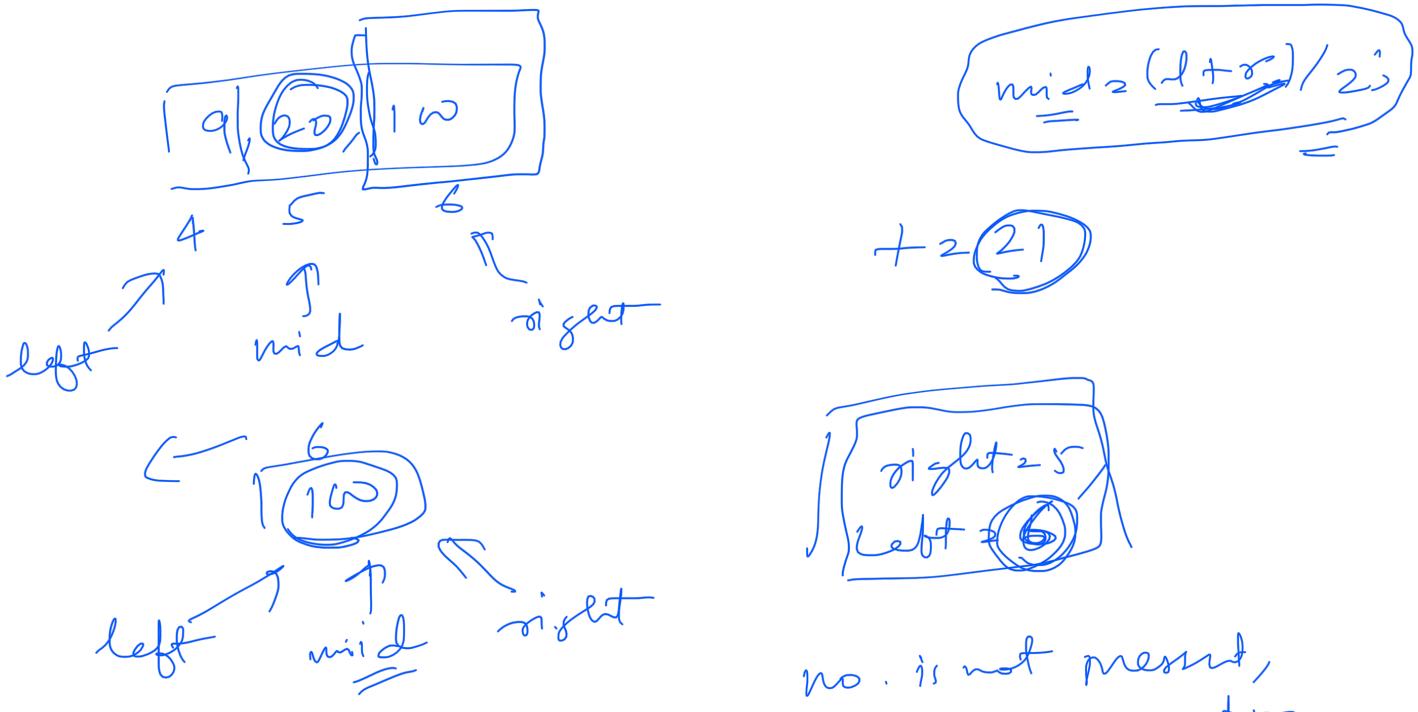
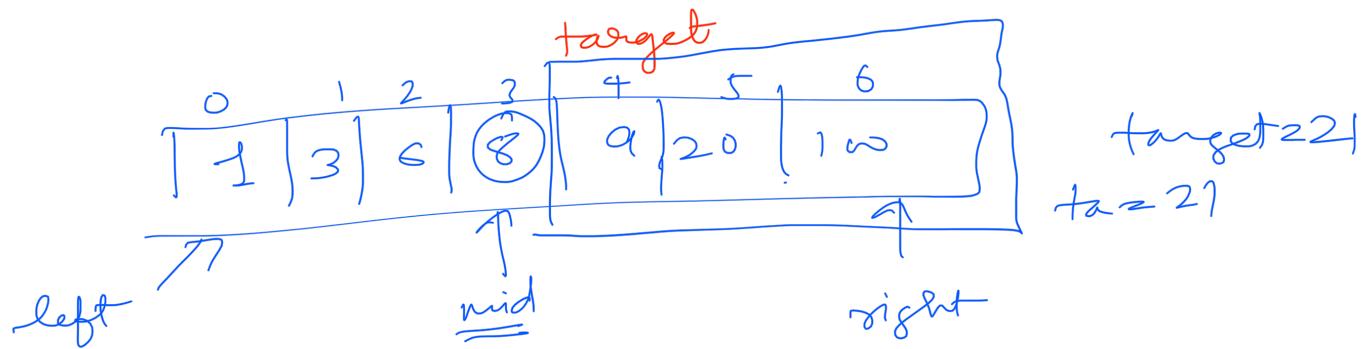
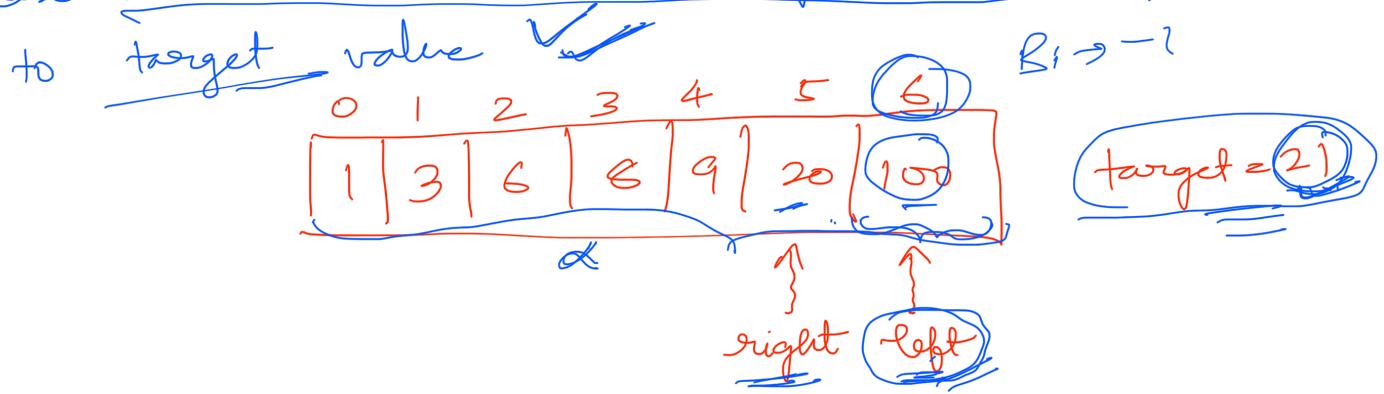
Can we use Binary search?

Binary search -



④ Solution =

1. Case when element in an array exists which is equal to target value ✓



$$\text{mid} = \frac{l+r}{2}$$

no. is not present,
we can return
left is

⑤ Time complexity / Space Complexity -

$$T = O(\log n)$$

$$S = O(1)$$

⑥ Code walkthrough -

```
int searchInsert (int [] nums, int target) {
```

$$\underline{\text{int left}} = 0;$$

$$\underline{\text{right}} = \underline{\text{nums.length}} - 1;$$

```
while ( $\underline{\text{left}} <= \underline{\text{right}}$ ) {
```

$$\underline{\text{int mid}} = \underline{\text{left}} + (\underline{\text{right}} - \underline{\text{left}}) / 2$$



```
if ( $\underline{\text{target}} \geq \underline{\text{nums}}[\underline{\text{mid}}]$ ) {
```

```
    return mid;
```

```
    } else if ( $\underline{\text{target}} > \underline{\text{nums}}[\underline{\text{mid}}]$ ) {
```

```
        left = mid + 1;
```

```
    } else {
```

```
        right = mid - 1;
```

$$\underline{\text{left}} = \frac{\text{MAX}}{2}$$

$$\frac{3M}{4}$$



$$\underline{\text{right}} = \frac{3\text{MAX}}{4} \rightarrow \text{INT_MAX}$$

$$= \frac{(\underline{\text{left}} + \underline{\text{right}})}{2}$$

$$\underline{\text{left}} + \underline{\text{right}} / 2$$

$$\left\{ \frac{M}{2} + \frac{3M}{4} \right\} / 2 \rightarrow \frac{2M + 3M}{4} = \frac{5M}{4}$$

$$\frac{2m+3m}{4} \rightarrow \frac{5m}{4}$$



```
return left;
```

```
left > right
```

```
nums[right] < target < nums[left]
```



```
left > right
```

```
nums[right] < target < nums[left]
```