In the end, it's YOU vs YOU. Be the best today, to defeat your yesterday!!

Binary Search

To find a given value in sorted search space

1	4	5	6	7	8	9	10	11	12
0	1	2	3	4	5	6	7	8	9

$x = 9$

e s

$s = 0$
$e = 9$
mid = 4

$s = 5$
$e = 9$
mid = 7

$s = 5$
$e = 6$
mid = 5

$s = mid + 1 = 6$
$e = 6$
$m = 6$

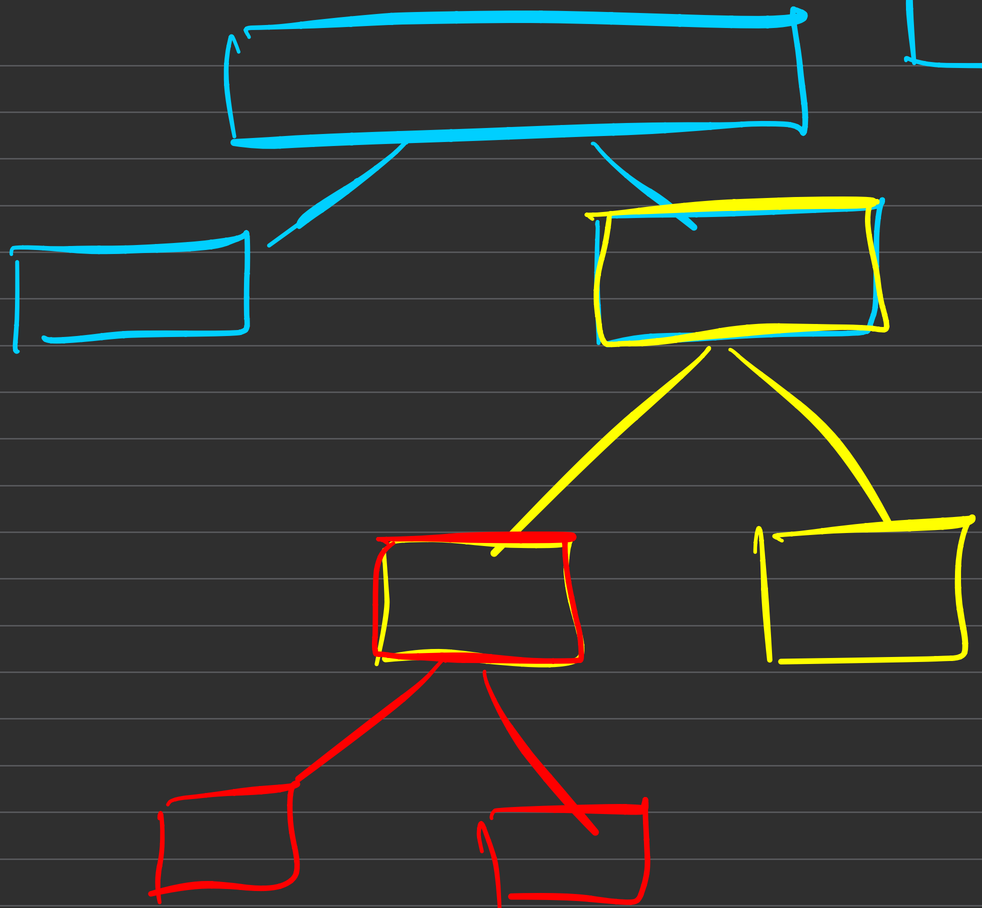
$9 = 9 = 6$

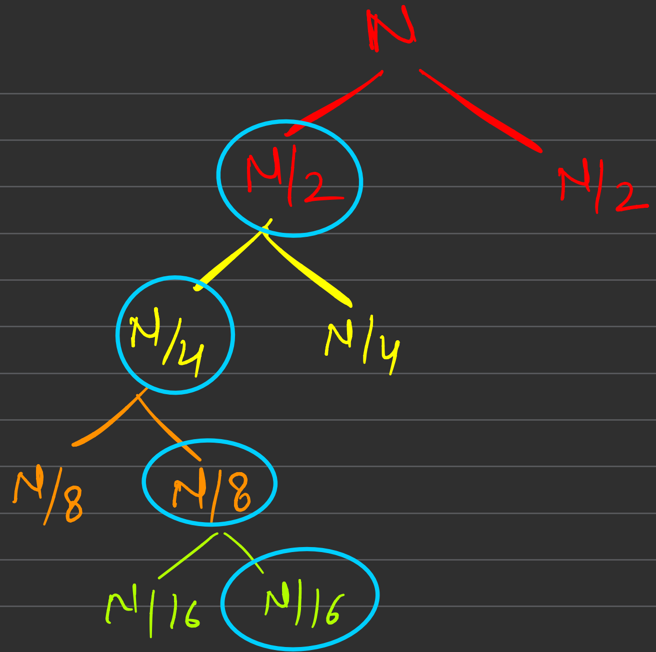
```
void binarySearch (arr, n, x)
{
    int start = 0 , end = n-1
    while ( start <= end )
    {
        mid = ( start + end ) /2
        if (arr[mid] == x) return mid

        if ( a[mid] < x )
            start = mid+1
        else
            end = mid-1
    }

    return -1
}
```

T:C: $O(\log N)$





$$N, \quad \frac{N}{2}, \quad \frac{N}{2^2}, \quad \frac{N}{2^3}, \quad \frac{N}{2^4}, \quad \ldots, \quad \frac{N}{2^K}$$

$$\frac{N}{2^K} = 1$$

$$N = 2^K$$

$$\log N = \log 2^K$$

$$\log N = K \log 2$$

$$\frac{\log N}{\log 2} = K$$

$$\boxed{K = \log_2 N}$$

Lower Bound

1	4	5	6	7	8	9	14	16	18
0	1	2	3	4	5	6	7	8	9

$N = 10$

$x = 4$

```
void LB(a, n, x)
{
    s=0 , e=n-1 , ans = N
    while (s <= e)
    {
        m = (s+e)/2
        if (a[mid] >= x)
        {   ans=mid
            e=mid-1
        }
        else{
            s=mid+1
}   }
```

return ans

$N = 5$

1	2	3	4	5
0	1	2	3	4

$\boxed{x = 9}$

$s = 0$
$e = 4$
$m = 2$

$s = 4$
$e = 4$
$m = 4$

$a(mid) \geq x$

ans = 5

$s = 3$
$e = 4$
$m = 3$

$s = 5$
$e = 4$

9

Upper Bound

0	1	2	3	4
1	2	6	7	8

$LB = 2 \Rightarrow 1$

$LB = 3 \Rightarrow 2$

$LB = 9 \Rightarrow 5$

$UB = 2 \Rightarrow 2$

$UB = 3 \Rightarrow 2$

$UB = 9 \Rightarrow 5$

35. Search Insert Position

Solved

Easy Topics Companies Re-do

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)` runtime complexity.

Example 1:

```
Input: nums = [1,3,5,6], target = 5
Output: 2
```

Example 2:

```
Input: nums = [1,3,5,6], target = 2
Output: 1
```

Example 3:

```
Input: nums = [1,3,5,6], target = 7
Output: 4
```

Floor in a Sorted Array

Difficulty: **Easy** Accuracy: **33.75%** Submissions: **475K+** Points: **2** Average Time: **30m**

Given a sorted array **arr[]** and an integer **x**, find the index (0-based) of the largest element in arr[] that is less than or equal to x. This element is called the **floor** of x. If such an element does not exist, return -1.

Note: In case of multiple occurrences of ceil of x, return the index of the last occurrence.

Examples

Input: arr[] = [1, 2, 8, 10, 10, 12, 19], x = 5
Output: 1
Explanation: Largest number less than or equal to 5 is 2, whose index is 1.

Input: arr[] = [1, 2, 8, 10, 10, 12, 19], x = 11
Output: 4
Explanation: Largest Number less than or equal to 11 is 10, whose indices are 3 and 4. The index of last occurrence is 4.

Input: arr[] = [1, 2, 8, 10, 10, 12, 19], x = 0
Output: -1
Explanation: No element less than or equal to 0 is found. So, output is -1.

1	2	4
0	1	2

$x = 0$

$ans = -1$

$s = 0$

$e = 2$

$m = 1$

$s = 0$

$e = 0$

$n =$

$s = 0$

$e = -1$

M