

December 16, 2023 5:43 PM

Example 1:
Input: nums = [-1,0,3,5,9,12], target = 9
Output: 4
Explanation: 9 exists in nums and its index is 4

Example 2:
Input: nums = [-1,0,3,5,9,12], target = 2
Output: -1
Explanation: 2 does not exist in nums so return -1

From <https://leetcode.com/problems/binary-search/>

```
Version 1 : [left, right]
Left = 0
Right = numSize - 1
While(left <= right):
    Middle = (left + right) / 2
    If(nums[middle] > target):
        Right = middle - 1
    Else if (nums[middle] < target):
        Left = middle + 1
    Else return middle
Return -1
```

```
Version 2: (left, right)
Left = 0
Right = numSize - 1
While(left < right):
    Middle = (left + right) / 2
    If(nums[middle] > target):
        Right = middle
    Else if (nums[middle] < target):
        Left = middle + 1
    Else return middle
Return -1
```

Custom Judge:

The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int val = ...; // Value to remove
int[] expectedNums = [...]; // The expected answer with correct length.
// It is sorted with no values equaling val.
int k = removeElement(nums, val); // Calls your implementation
assert k == expectedNums.length;
sort(nums, 0, k); // Sort the first k elements of nums
for (int i = 0; i < actualLength; i++) {
    assert nums[i] == expectedNums[i];
}
```

If all assertions pass, then your solution will be **accepted**.

Constraints:

- `0 <= nums.length <= 100`
- `0 <= nums[i] <= 50`
- `0 <= val <= 100`

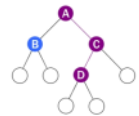
From <https://leetcode.com/problems/remove-element/>

A tree is ordered if there is a meaningful linear order among the children
(Left to Right)

```

graph TD
    A[Want a fast meal?] -- Yes --> B[How about coffee?]
    A -- No --> C[On expense account?]
    B -- Yes --> D[Starbucks]
    B -- No --> E[Chipotle]
    C -- Yes --> F[Grace's]
    C -- No --> G[Café Paragon]
  
```

Example :



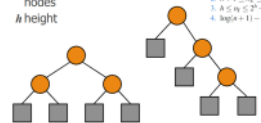
node A:
Height = 3 (no. of edges on longest path from A is A \rightarrow C \rightarrow D \rightarrow leaf i.e. 3 edges)
depth = 0

Notation

- n number of nodes
- e number of external nodes
- i number of internal nodes
- h height

Also, if T is proper, then T has the following properties

- $1. h+1 \leq n \leq 2^{h+1} - 1$
- $1 \leq n_e \leq 2^h$
- $h \leq n_i \leq 2^h - 1$
- $\log(n+1) - 1 \leq h \leq n-1$
- $1. 2h+1 \leq n \leq 2^{h+1} - 1$
- $2. h+1 \leq n_e \leq 2^h$
- $3. h \leq n_i \leq 2^h - 1$
- $4. \log(n+1) - 1 \leq h \leq (n-1)/2$



Return slow