

Easy

You are given an array of **distinct** integers `nums`, sorted in ascending order, and an integer `target`.

Implement a function to search for `target` within `nums`. If it exists, then return its index, otherwise, return `-1`.

Your solution must run in $O(\log n)$ time.

Example 1:

Input: `nums = [-1,0,2,4,6,8]`, `target = 4`

Output: `3`

Example 2:

Input: `nums = [-1,0,2,4,6,8]`, `target = 3`

Output: `-1`

Constraints:

- `1 <= nums.length <= 10000`.
- `-10000 < nums[i]`, `target < 10000`

M
R L

-1 0 2 4 6 8
target = 10

Left = 0

Right = len(nums) - 1

while (L <= R)

 mid = (L + R) // 2

 // floor and ceiling division

 if target > nums[M]

 L = M + 1

 else if target < nums[M]

 R = M - 1

 else

 return m

return -1