704. Binary Search

Given an array of integers `nums` which is sorted in ascending order, and an integer `target`, write a function to search `target` in `nums`. If `target` exists, then return its index. Otherwise, return `-1`.

You must write an algorithm with `O(log n)` runtime complexity.

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

```
def search(self, nums: List[int], target: int) -> int:
        if nums[0]==target:
            return 0
        if nums[-1]==target:
            return len(nums)-1


        lo=0
        hi = len(nums)-1
        while lo<=hi:
            mid = (lo+hi)//2
            if nums[mid]==target:
                return mid
            elif nums[mid]>target:
                hi = mid-1
            else:
                lo = mid+1
        return -1
```