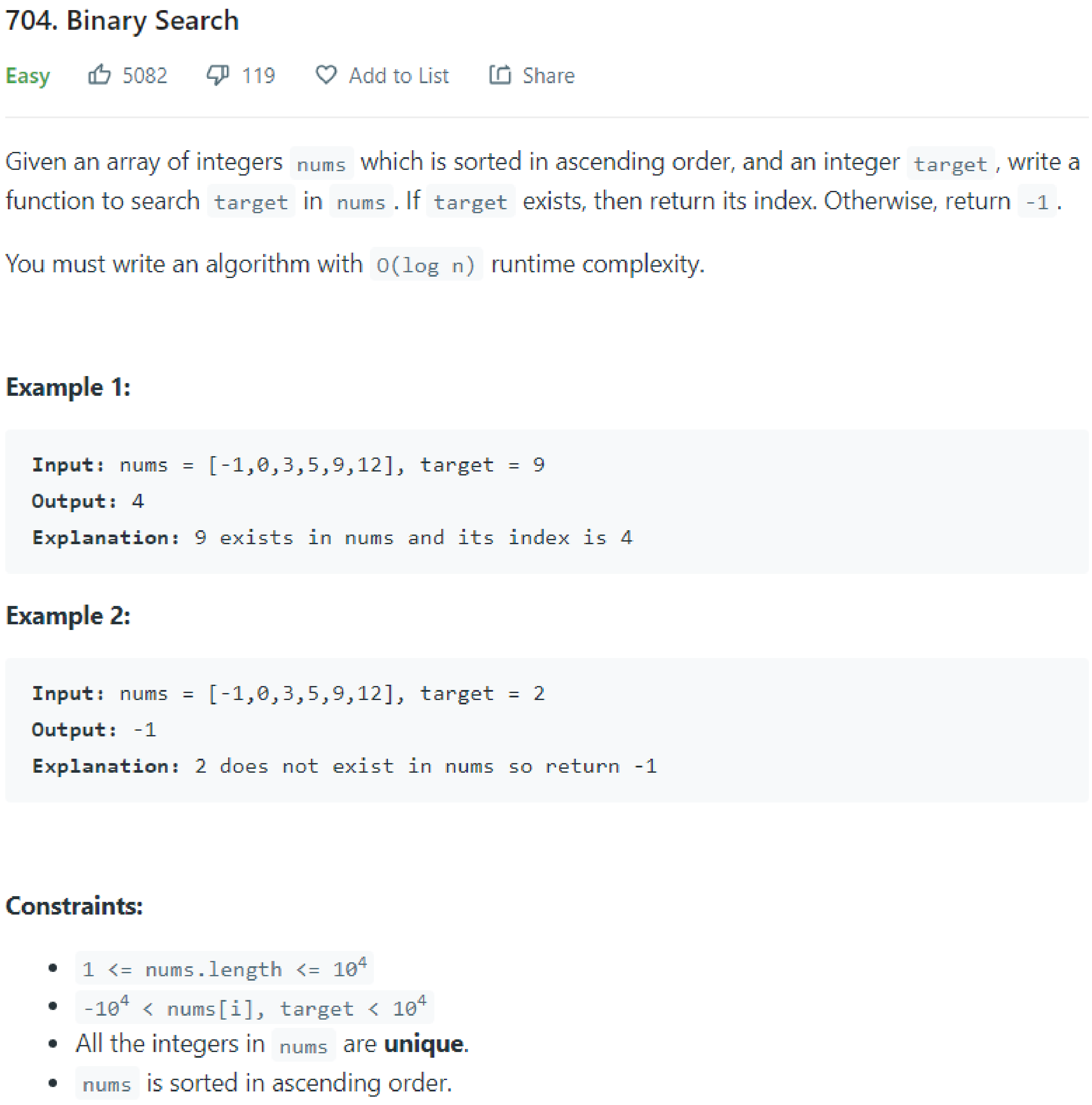


704. Binary Search

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:

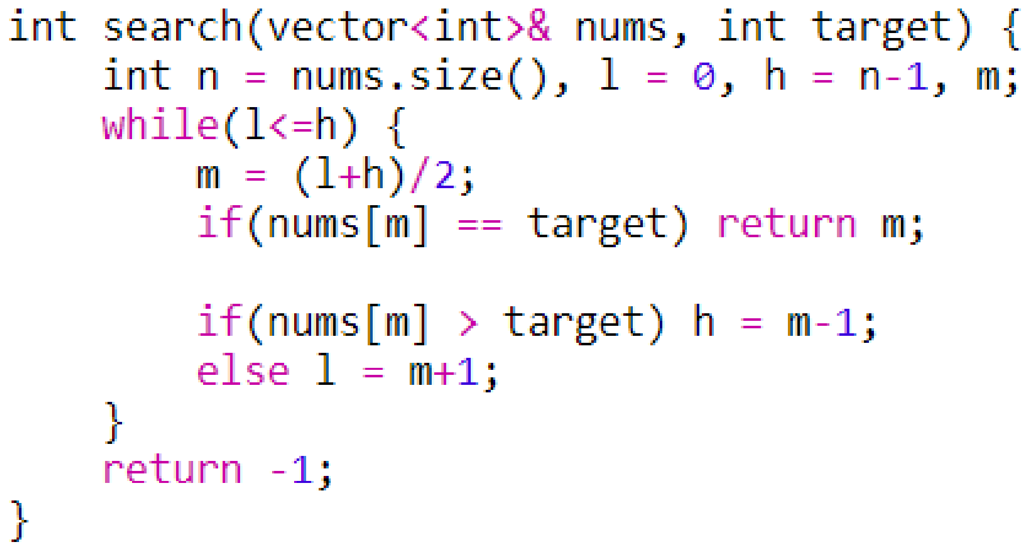
```
Input: nums = [-1,0,3,5,9,12], target = 9
Output: 4
Explanation: 9 exists in nums and its index is 4
```

Example 2:

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

Constraints:

- $1 <= \texttt{nums.length} <= 10^4$
- $-10^4 < \texttt{nums[i]}, \texttt{target} < 10^4$
- All the integers in `nums` are **unique**.
- `nums` is sorted in ascending order.



```
int search(vector<int>& nums, int target) {
    int n = nums.size(), l = 0, h = n-1, m;
    while(l<=h) {
        m = (l+h)/2;
        if(nums[m] == target) return m;

        if(nums[m] > target) h = m-1;
        else l = m+1;
    }
    return -1;
}
```

#100daysofDSA

/rvislive

Rakesh Vishwakarma