704. Binary Search

Tags	
Property	@September 24, 2022

Question

原文：

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.

我的理解：

給定一個已排序好的陣列跟目標，回傳目標在陣列中的位置，找不到的話就回傳-1

翻譯：

自評翻譯正確性：

- Word Memory：

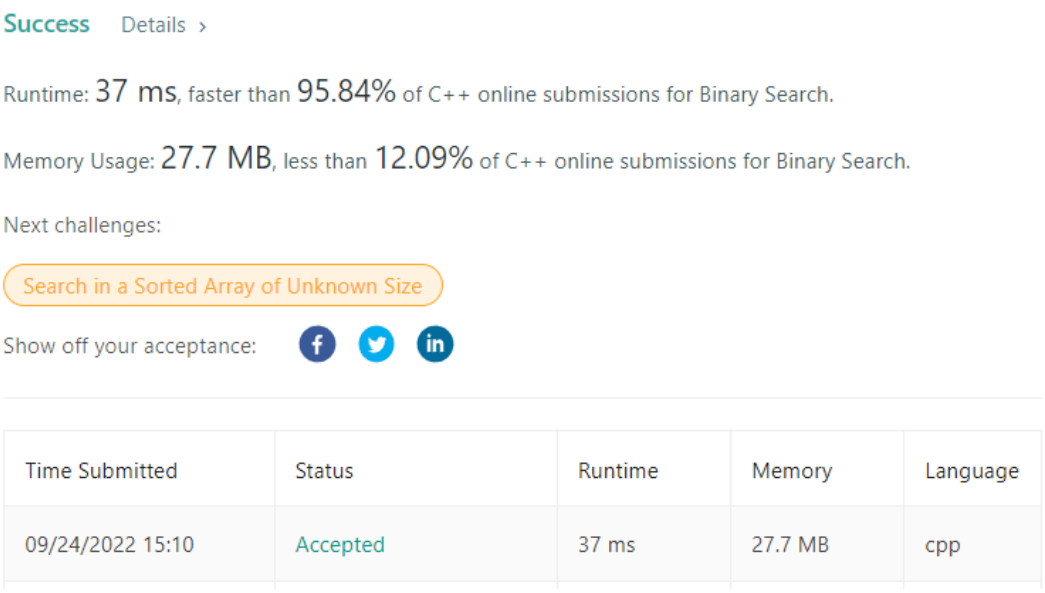
Code

```
class Solution {
public:
    int search(vector<int>& nums, int target) {
        int left=0,right=nums.size(),mid;

        while(left<right){
            mid=(left+right)/2;
            if(nums[mid]==target){
                return mid;
            }
            else if(mid==left){//如果mid在計算後等同left表示left、right已經是相鄰的，爾且又沒找到Target 之後的計算也會是無限輪迴 所以要break
                break;
            }
            else if(nums[mid]<target){
                left=mid;
            }
            else if(nums[mid]>target){
                right=mid;
            }
        }

        return -1;
    }
};
```

> 思路：就是二元搜尋的思考邏輯



優良code參考

```
int search(vector<int>& nums, int target) {
        int n = nums.size()-1;
        int low = 0, high = n;
        while( low <= high){
            int mid = low + (high-low)/2;
            if (nums[mid] == target) return mid;
            else if (nums[mid] > target) high = mid -1;
            else low = mid + 1;
        }
        return -1;
  }
```

> 思路：