

# 034. Search for a Range

## 034 Search for a Range

- Binary Search+array

### Description

Given an array of integers sorted in ascending order, find the starting and ending position of a given target value.

Your algorithm's runtime complexity must be in the order of  $O(\log n)$ .

If the target is not found in the array, return `[-1, -1]`.

For example,

Given `[5, 7, 7, 8, 8, 10]` and target value 8,  
return `[3, 4]`.

### 1. Thought line

### 2. Binary Search+array

```
1 class Solution {
2 private:
3     void binarySearchRange(vector<int>& nums, int target, int st, int ed, int& indexMin, int& indexMax){
4         if (st>ed) return;
5         if (target<nums[st] || target>nums[ed]) return;
6         if (st==ed && nums[st]!=target) return;
7         if (st==ed && nums[st] == target){
8             if (indexMin == -1 && indexMax == -1)
9                 indexMin = indexMax = st;
10            else{
11                indexMin = st<indexMin?st:indexMin;
12                indexMax = st>indexMax?st:indexMax;
13            }
14        }
15        else{
16            int mid = (st+ed)/2;
17            binarySearchRange(nums, target, st, mid, indexMin, indexMax);
18            binarySearchRange(nums, target, mid+1, ed, indexMin, indexMax);
19        }
20    }
21
22 public:
23     vector<int> searchRange(vector<int>& nums, int target) {
24         int indexMin = -1, indexMax = -1;
25         binarySearchRange(nums, target, 0, nums.size()-1, indexMin, indexMax);
26         return vector<int>{indexMin, indexMax};
27     }
28 };
```