34. Search for a Range

题目描述https://leetcode.com/problems/search-for-a-range/

> 给定一个按序号排列的数组，和一个target，找到target在数组中的位置的范围。如果没有返回[-1, -1]
> 例如：
>
> ```
> [5, 7, 7, 8, 8, 10] and target value 8
> return [3, 4].
> [1,1,1,1] and target 1
>   return [0, 3]
> [1,1,1,1] and target 0
> return[-1, -1]
> ```

解题思路：

1. 遍历
2. 用binary search首先找到左边界限，然后找右边界限。

代码1：

```
class Solution {
public:
    vector<int> searchRange(vector<int>& nums, int target) {
        int b = -1, e = -1;
        vector<int> res(2, -1);
        for(int i = 0; i < nums.size(); i++){
            b = i;
            while(i<nums.size() && nums[i] == target){
                i++;
            }
            e = i-1;
            if(b <= e){
                res[0] = b;
                res[1] = e;
                return res;
            }
        }
        return res;
    }
};
```

代码2：

```
class Solution {
public:
    vector<int> searchRange(vector<int>& nums, int target) {
        int len = nums.size();
        int l = 0, r = len - 1, mid = 0;
        int left = 0, right = 0;
        //find Left
        while(l < r) {
            mid = l + (r - l) / 2;
            if(nums[mid] < target) {
                l = mid + 1;
            }
            else {
                r = mid;
            }
        }
        if(nums[l] != target) {
            return {-1,-1};
        }
        left = l;
        l = left; r = len - 1;
        cout << left << endl;
        while(l < r) {
            mid = l + (r - l) / 2 + 1;
            if(nums[mid] <= target) {
                l = mid;
            }
            else {
                r = mid - 1;
            }
        }
        right = l;
        return {left, right};
    }
};
```