

34. 在排序数组中查找元素的第一个和最后一个位置

二分查找不仅仅能用在有序的数组中，如果数组有明显的分区，例如左面比target小，右面比target大，这种分区，依然可以使用二分查找

△

1. 查找区间的左端点

细节处理:

1. 循环条件

$left \leq right$ ✗

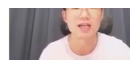
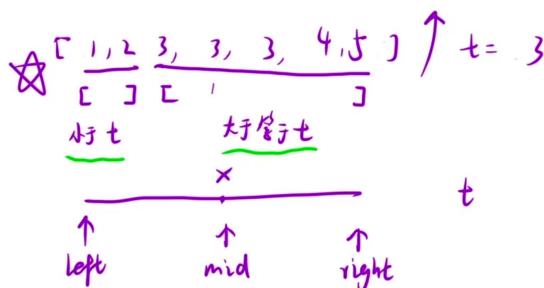
$left < right$ ✓

- △ {
1. $left = right$ 的时候，就是最终结果，无需判断
 2. 如果判断，就会死循环

△△

2. 求中点的操作

① $left + (right - left) / 2$ ✓



- ☆ {
- ① $x < t \rightarrow left = mid + 1 \rightarrow [left, right]$
 - ② $x \geq t \rightarrow right = mid \rightarrow [left, right]$

△

△

mid

↓

00

↑

↑

① 有结果

② 全大于 t

ret

↑

right

```
1 class Solution {
2 public:
3     vector<int> searchRange(vector<int>& nums, int target) {
4         if(nums.empty())
5         {
6             return {-1,-1};
7         }
8         int n = nums.size();
9         vector<int> ret;
10        int left = 0;
11        int right = n-1;
12
13        //左端点
14        while(left < right)
15        {
16            int mid = left + (right-left)/2;
17            if(nums[mid] < target) left = mid+1;
18            else right = mid;
19        }
20        if(nums[left] != target) return {-1,-1};
21        else ret.push_back(left);
22
23        left = 0;
24        right = n-1;
25        while(left < right)
26        {
27            int mid = left + (right-left+1)/2;
28            if(nums[mid] > target) right = mid-1;
29            else left = mid;
30        }
31        ret.push_back(left);
32        return ret;
33    }
34 };
```