**描述**

给定一个排序的整数数组（升序）和一个要查找的整数target，用O(logn)的时间查找到target第一次出现的下标（从0开始），如果target不存在于数组中，返回-1。

因为是要返回第一个下标，所以的话还得不断缩小右边界。整体的思路就是二分查找的思路了。需要注意的细节就是右边界的确定。如果是size()-1，循环条件是left<=right,否则是left<right。

以下是代码：

int binarySearch(vector<int> &nums, int target) {

// write your code here

int right=nums.size()-1,left=0;

while(left<=right){

int mid=left+(right-left)/2;

if(nums[mid]<target)

left=mid+1;

else if(nums[mid]>target)

right=mid-1;

else if(nums[mid]==target)

right=mid-1;

}

if(left>=nums.size()||nums[left]!=target)

return -1;

return left;

}

截图如下：

