在排序数组中查找元素的第一个和最后一个位置

给定一个按照升序排列的整数数组 nums，和一个目标值 target。找出给定目标值在数组中的开始位置和结束位置。

如果数组中不存在目标值 target，返回 [-1, -1]。

进阶：

- 你可以设计并实现时间复杂度为 O(log n) 的算法解决此问题吗？

示例 1：

输入： nums = [5,7,7,8,8,10], target = 8

输出： [3,4]

示例 2：

输入： nums = [5,7,7,8,8,10], target = 6

输出： [-1,-1]

示例 3：

输入： nums = [], target = 0

输出： [-1,-1]

```
/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
 int indexsearch(int* nums, int numsSize, int target,bool flag)
 {
     int left=0;
     int right=numsSize-1;
     int ans=numsSize;
     while(left<=right)
     {
         int mid=left+(right-left)/2;
         if(nums[mid]>target||(flag&&nums[mid]>=target))
           {
                right=mid-1;
                ans=mid;
           }
         else
```

```
            left=mid+1;
        }
        return ans;
    }
int* searchRange(int* nums, int numsSize, int target, int* returnSize){
    //二分查找
    int startIdx = indexsearch(nums, numsSize, target,true);
    int endIdx = indexsearch(nums, numsSize, target,false)-1;
    int* ret = malloc(sizeof(int) * 2);
    *returnSize = 2;
    if (startIdx <= endIdx && endIdx < numsSize && nums[startIdx] == target
 && nums[endIdx] == target) {
        ret[0] = startIdx, ret[1] = endIdx;
        return ret;
    }
    ret[0] = -1, ret[1] = -1;
    return ret;
}
```