至少是其他数字两倍的最大数

在一个给定的数组 nums 中，总是存在一个最大元素 。

查找数组中的最大元素是否至少是数组中每个其他数字的两倍。

如果是，则返回最大元素的索引，否则返回-1。

示例 1:

输入: nums = [3, 6, 1, 0]

输出: 1

解释: 6 是最大的整数，对于数组中的其他整数，

6 大于数组中其他元素的两倍。6 的索引是 1，所以我们返回 1.

示例 2:

输入: nums = [1, 2, 3, 4]

输出: -1

解释: 4 没有超过 3 的两倍大，所以我们返回 -1.

解题思路：

先排序，然后以当前末尾最大值仅与前一元素（第二大）比较即可判断 返回对应原数组下标

```
int cmp(const void* a,const void* b)
{
    return(*(int*)a-*(int*)b);
}
int dominantIndex(int* nums, int numsSize){
    if(numsSize==1)
        return 0;
    int arr[numsSize];
    for (int i = 0; i < numsSize; i++) {
        arr[i] = nums[i];
    }
    qsort(nums,numsSize,sizeof(int),cmp);
    if(nums[numsSize-1]<2*nums[numsSize-2])
        return -1;
```

```
    for(int i=0;i<numsSize;i++)
    {
        if(arr[i]==nums[numsSize-1])
            return i;
    }
    return -1;
}
```