

最大子序和

给定一个整数数组 `nums`，找到一个具有最大和的连续子数组（子数组最少包含一个元素），返回其最大和。

示例 1:

输入: `nums = [-2,1,-3,4,-1,2,1,-5,4]`

输出: 6

解释: 连续子数组 `[4,-1,2,1]` 的和最大, 为 6。

示例 2:

输入: `nums = [1]`

输出: 1

示例 3:

输入: `nums = [0]`

输出: 0

示例 4:

输入: `nums = [-1]`

输出: -1

示例 5:

输入: `nums = [-100000]`

输出: -100000

//滑窗思想

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        if(nums.size()==0)
        {
            return 0;
        }
        int left=0;//滑窗 左边界索引
```

```
int right=0;//滑窗 右边界索引

int cursum=nums[0];//记录当前滑窗和
int maxsum=nums[0];

while(left<=right&&right<nums.size()-1)
{
    right++;
    if(cursum<0&&cursum<nums[right])
    {
        //当前滑窗和<滑窗右边界单值 或者当前滑窗为负数 需要更新左边界 重新确定滑窗
        left=right;
        cursum=0;//当前滑窗和 归零
    }
    cursum+=nums[right];
    maxsum=maxsum>cursum?maxsum:cursum;//根据大小关系 关系 maxsum
}
return maxsum;
}
};
```