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:

输入： `[-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`。

进阶:

如果你已经实现复杂度为 O(*n*) 的解法，尝试使用更为精妙的分治法求解。

思路，和股票一样，进行贪婪算法的动态生成，需要注意，可能存在没有正数的情况，这时就需要选择最大的数即可

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int num=0;
        int max=0;
        int min=1;
        int n=nums.size();
        for(int i=0;i<n;++i){
            if(nums[i]<=0){
                if(min==1){
                    min=nums[i];
                }else{
                    min=min>nums[i]?min:nums[i];
                }
            }
            if(num>0){
                num+=nums[i];
                max=max>num?max:num;
                if(num < 0){
                    num=0;
                }
            }else if(nums[i]>0){
                num+=nums[i];
```

```
                max=max>num?max:num;
            }
        }
        if(max==0){
            return min;
        }
        return max;
    }
};
```