/*Find the contiguous subarray within an array (containing at least one number) which has the largest sum.

For example, given the array [-2,1,-3,4,-1,2,1,-5,4], the contiguous subarray [4,-1,2,1] has the largest sum = 6.*/

-
- 思想：

- (1) 分治思想：只有一个数的时候，答案就是它；有2个数的时候，如果第1个数大于0，答案是Math.max(nums[0], nums[0]+nums[1])，如果第1个数小于0，答案是 Math.max(nums[0], nums[1])；当有n个数的时候，现在已知 前n-1个数的答案，那么考虑第n个数时，答案在 Math.max(f(n-1), 包含n的某子数列),其中包含n的某子数列分为两种情况：如果n之前的子数列小于0了，那么直接就是n，如果n之前的子数列大于0，那么是包含n的子数列

```
public int maxSubArray(int[] nums) {
    if (nums == null || nums.length == 0)  return 0;
    int sum=nums[0];
    int max=nums[0];
    for(int i=1;i<nums.length;i++){
        if(sum<0) sum=nums[i];
        else sum+=nums[i];
        max=Math.max(max,sum);
    }
    return max;
}
```