# [最大子序和](https://leetcode-cn.com/explore/interview/card/bytedance/246/dynamic-programming-or-greedy/1029/)

**头条重点**

## 题目

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。

## 解题思路

1. 动态规划：

$$f(i)=\left\{\begin{matrix}num[i]&f(i−1)+num[i]<num[i]\\f(i−1)+num[i]&f(i−1)+num[i]>num[i]\end{matrix}\right.$$

1. 用result[i]保存以数字nums[i]结尾的最大子序和，然后不断更新result数组的最大值即可。总的时间复杂度O(n)

public int maxSubArray(int[] nums) {
 if (nums.length == 0) {
 return 0;
 }

 if (nums.length == 1) {
 return nums[0];
 }

 int[] res = new int[nums.length];
 res[0] = nums[0];

 int max = res[0];
 for (int i = 1; i < nums.length; i++) {
 int curMax = nums[i] + res[i - 1];
 if (curMax > nums[i]) {
 res[i] = curMax;
 } else {
 res[i] = nums[i];
 }
 max = Math.max(max, res[i]);
 }

 return max;
}