

41. 最大子数组

给定一个整数数组，找到一个具有最大和的子数组，返回其最大和。

子数组最少包含一个数

给出数组 `[-2, 2, -3, 4, -1, 2, 1, -5, 3]`，符合要求的子数组为 `[4, -1, 2, 1]`，其最大和为 `6`

nums: `[-2, 1, -3, 4, -1, 2, 1, -5, 4]`

状态f: `[-2, 1, -2, 4, 3, 5, 6, 1, 5]`

状态：f[i]代表 `maxSubArray(0...i)`，即前面包括i的最大子数组和

状态转移： $f[i] = f[i-1] > 0 ? \text{nums}[i] + f[i-1] : \text{nums}[i]$ 。如果f[i-1]为负，则f[i]为当前nums[i]，因为如果加上f[i-1]的话，会让和变得更小；只有f[i-1]>0时，加上f[i-1]才会让和更大。

```
1 public int maxSubArray(int[] nums) {
2     if( null == nums || nums.length == 0 )
3         return 0;
4     int[] sums = new int[nums.length];
5     sums[0] = nums[0];
6     int ans = nums[0];
7     for( int i = 1; i < nums.length; i++ ){
8         if( sums[i-1] < 0 )
9             sums[i] = nums[i];
10        else
11            sums[i] = nums[i] + sums[i-1];
12        ans = Math.max(ans, sums[i]);
13    }
14    return ans;
15 }
```

本题可以不用数组。因为只用到了sums[i]和sums[i-1]这两个数，因为sums[i]是根据sums[i-1]更新的，所以可以只用一个数sum来替代。