**难度: 简单**

给定一个整数数组，找出总和最大的连续数列，并返回总和。

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。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/contiguous-sequence-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路:**

1). 每次把下标的数进行相加

2). 如果加完之后小于原本的数,那么将那些数舍弃,直接让这个数,当成第一个

|  |
| --- |
| public int maxSubArray(int[] nums) {  int sum = 0;  int max = Integer.MIN\_VALUE; //最大值  int index = 0; //下标  while (index < nums.length) {  //和数 加 上当前下标的数 小于 当前下标的数  if (sum + nums[index] < nums[index]) {  //置为0  sum = 0;  }    //相加  sum += nums[index];  if (sum > max) {  //如果和数大于最大值,互换  max = sum;  }  //下标移动1  index++;  }  return max;  } |