



链滴

LeetCode 53. 最大子序和 (java 解法)

作者: [wky181](#)

原文链接: <https://ld246.com/article/1569721648965>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



最大子序和

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），返回其大和。

示例 1:

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。

这道题最简单的方法就是暴力破解，找出具有最大和的连续子数组，用二重for循环来做，话不多说先上这部分代码：

```
public static int maxSubArray1(int[] nums) {
    int num = 0;
    int maxNums = Integer.MIN_VALUE;
    for (int i = 0; i < nums.length; i++) {
        num = 0;
        //以下标i开头，下标j结尾，找出其中连续子数组的最大和
        for (int j = i; j < nums.length; j++) {
            num += nums[j];
            if (num > maxNums) {
                maxNums = num;
            }
        }
    }
    return maxNums;
}
```

用双重for循环找出了所有连续数组的情况，算出每种连续数组情况中连续子数组和的最大值，并比较出最终结果。这种方法是比较好想的，但是时间复杂度是n的2次方。

其实这样的时间复杂度不是很理想，n的平方级的复杂度已经是比较高的了，可以再进行优化。用动规划的思想来做这道题。

- 当前最大连续子序列和为 currntNums，结果为 res
- 如果 currntNums > 0，则说明 currntNums 对结果有增益效果，则 currntNums 保留并加上当前遍历数字
- 如果 currntNums <= 0，则说明 currntNums 对结果无增益效果，需要舍弃，则currntNums 直更新为当前遍历数字
- 每次比较 currntNums和 res 的大小，将最大值置为res，遍历结束返回结果

```
public static int method (int[] nums){
    int res =nums[0];
    //当前最大连续序列的和（不是最大和）
    int currntNums = 0;
    for (int i = 0; i < nums.length ; i++) {
        //考虑是否对结果是否有增益
        if (currntNums>0){
            currntNums+=nums[i];
        }else {
            currntNums = nums[i];
        }
        res = Math.max(currntNums,res);
    }
    return res;
}
```