

连续子数组的最大和

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518866757772>

来源网站: [链滴](#)

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

题目描述

HZ偶尔会拿些专业问题来忽悠那些非计算机专业的同学。今天测试组开完会后,他又发话了:在古老的维模式识别中,常常需要计算连续子向量的最大和,当向量全为正数的时候,问题很好解决。但是,如果向中包含负数,是否应该包含某个负数,并期望旁边的正数会弥补它呢? 例如:{6,-3,-2,7,-15,1,2,2},连续子向量的最大和为8(从第0个开始,到第3个为止)。你会不会被她忽悠住? (子向量的长度至少是1)

解题思路

动态规划。

- 先设定第一个数是临时最优解;
- 设定现在的子数组和为0, 因为还没有与任何数字相加;
- 当前的子数组与下一个数字相加:
 - 如果相加的和大于临时最优解, 则对临时最优解进行更新;
 - 如果现在的和已经小于0, 则不需要再保持现在的和, 将tmp=0;

代码

```
public class Solution {
    public int FindGreatestSumOfSubArray(int[] array) {
        if (array == null || array.length == 0)
            return 0;
        int ret = array[0];
        int tmp = 0;
        for (int i = 0; i < array.length; i++) {
            tmp += array[i];
            ret = ret > tmp ? ret : tmp;
            if (tmp < 0)
                tmp = 0;
        }
        return ret;
    }
}
```