题目描述：

Find the contiguous subarray within an array (containing at least one number) which has the largest sum.

For example, given the array `[-2,1,-3,4,-1,2,1,-5,4]`,

the contiguous subarray `[4,-1,2,1]` has the largest sum = `6`.

自己的解题思路：

分别寻找 [0,len-1],[1,len-1],[2,len-1],.....[len-1,len-1]的max value array，时间复杂度为o(n2) 结果：超时

题解思路：这是一个DP问题，先划分是子问题，子问题是求[0,i]与[0,i-1]的最大和

题解代码：

```
public int maxSubArray(int[] A) {
        int n = A.length;
        int[] dp = new int[n];//dp[i] means the maximum subarray ending with A[i];
        dp[0] = A[0];
        int max = dp[0];

        for(int i = 1; i < n; i++){
            dp[i] = A[i] + (dp[i - 1] > 0 ? dp[i - 1] : 0);
            max = Math.max(max, dp[i]);
        }

        return max;
}
```