```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:45 CST 2018*/

/*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.*/


import java.util.*;
import java.lang.Math;
import java.lang.System;
import java.lang.Integer;


public class Main {

    public static void main(String[] args) throws ArithmeticException {

        int[] input = {7, 1, 5, 3, 6, 4};

        Solution solution = new Solution();

        int result = solution.maxProfit(input);

        System.out.println("haha");
    }

}


class ListNode {
    int val;
    ListNode next;

    ListNode(int x) {
        val = x;
    }
}


class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;

    TreeNode(int x) {
        val = x;
    }
}


class Solution {
    public int maxSubArray(int[] nums) {

        int numsLength = nums.length;
        int result = nums[0];
        int tempSum = 0;

        for (int i = 0; i < numsLength; i++) {

            result = Math.max(result, tempSum + nums[i]);
            tempSum = Math.max(tempSum + nums[i], 0);
        }

        return result;
    }
}
```