41. Maximum Subarray

Given an array of integers, find a contiguous subarray which has the largest sum.

Example

**Example1:**  
Given the array [−2,2,−3,4,−1,2,1,−5,3], the contiguous subarray [4,−1,2,1] has the largest sum = 6.  
**Example2:**  
Given the array [1,2,3,4], the contiguous subarray [1,2,3,4] has the largest sum = 10.

Challenge

Can you do it in time complexity O(n)?

Notice

The subarray should contain at least one number.

<https://www.lintcode.com/problem/maximum-subarray/description>

#include <iostream>

#include <stdio.h>

#include <vector>

using namespace std;

int maxSubArray(vector<int> &nums) {

// write your code here

int m = nums[0];

int actual = nums[0];

for(int i = 1; i < nums.size(); i++) {

//int actual = nums[i]

actual = max(nums[i], actual + nums[i]);

m = max(m, actual);

}

return m;

}

int main() {

vector<int> v;

v.push\_back(-1);

cout << maxSubArray (v) << endl;

return 0;

}