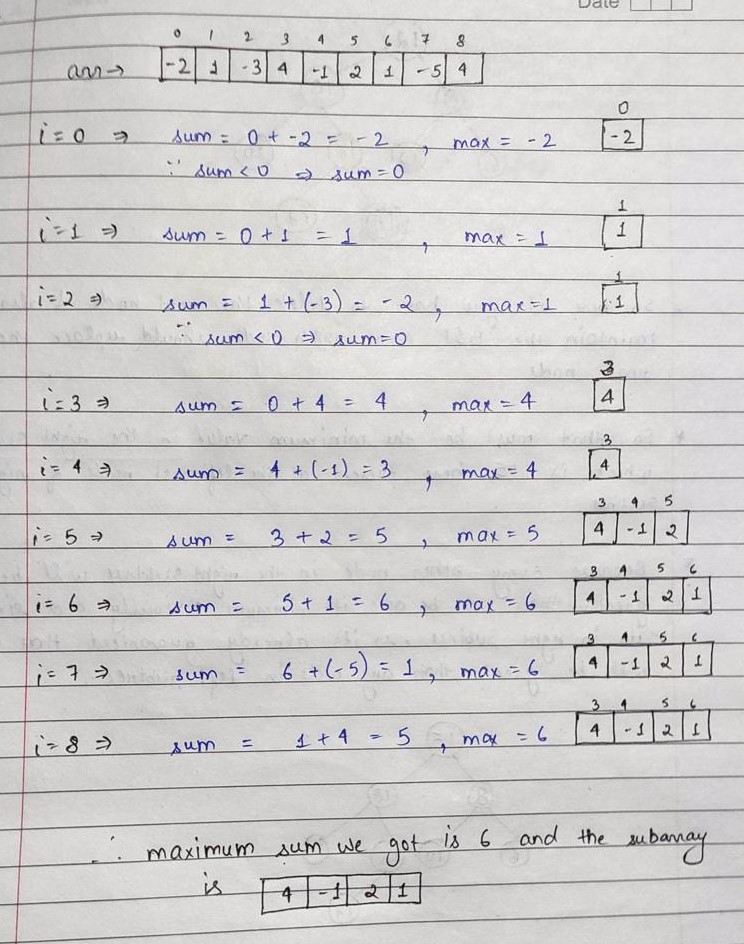
Question: <https://leetcode.com/problems/maximum-subarray/>

Start traversing your array keep each element in the sum and every time keep the max of currSum and prevSum.  
**But the catch here is that if at any point sum becomes negative then no point keeping it because 0 is obviously greater than negative, so just make your sum 0.**

Now here in this question you can see that you can also be asked some more things like :

* Length of the max subarray
* Elements of the max subarray
* Start and End index of max subarray



Code:  
class Solution {

public int maxSubArray(int[] nums) {

int max\_so\_far=Integer.MIN\_VALUE,max\_ending\_here=0;

if(nums.length==1){

return nums[0];

}

for(int i=0;i<nums.length; i++){

max\_ending\_here+=nums[i];

if(max\_so\_far<max\_ending\_here){

max\_so\_far=max\_ending\_here;

}

if(max\_ending\_here<0){

max\_ending\_here=0;

}

}

return max\_so\_far;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>