Maximum Subarray Sum:

Brute:

O(n2)->Time

def maxSubarraySum(arr, n) :

    maxi=0

    for i in range(n):

        sum1=0

        for j in range(i,n):

            sum1+=arr[j]

            maxi=max(sum1,maxi)

    return maxi

Optimal:

def maxSubarraySum(arr, n) :

    maxi=0

    sum1=0

    for i in range(n):

        sum1+=arr[i]

        if sum1<0:

            sum1=0

        maxi=max(sum1,maxi)

    return maxi

Follow up:

To find the subarray with maximum sum

def maximumSumSubarray(n, arr):

    maxi,sum1 = 0,0

    starti = 0

    end,start = 0,0

    sum1 = 0

    for i in range(n):

        sum1 += arr[i]

        if (sum1 > maxi):

            maxi = sum1

            starti = start

            end = i

        #return longer of the two subarrays

        elif(sum1==maxi):

            if(end-starti<i-start):

                starti=start

                end=i

        if sum1<0:#Update starting point when sum starts to go negative

            sum1=0

            start=i+1

    result = []

    if maxi==0:

        result.append(max(arr))

        return result

    for i in range(starti, end+1):

        result.append(arr[i])

    return result