Maximum Sum Circular Subarray

Solution

Given a **circular array C** of integers represented by `A`, find the maximum possible sum of a non-empty subarray of **C**.

Here, a *circular array* means the end of the array connects to the beginning of the array. (Formally, `C[i] = A[i]` when `0 <= i < A.length`, and `C[i+A.length] = C[i]` when `i >= 0`.)

Also, a subarray may only include each element of the fixed buffer `A` at most once. (Formally, for a subarray `C[i], C[i+1], ..., C[j]`, there does not exist `i <= k1, k2 <= j` with `k1 % A.length = k2 % A.length`.)

Example 1:

```
Input: [1,-2,3,-2]
Output: 3
Explanation: Subarray [3] has maximum sum 3
```

Example 2:

```
Input: [5,-3,5]
Output: 10
Explanation: Subarray [5,5] has maximum sum 5 + 5 = 10
```

Example 3:

```
Input: [3,-1,2,-1]
Output: 4
Explanation: Subarray [2,-1,3] has maximum sum 2 + (-1) + 3 = 4
```

Example 4:

```
Input: [3,-2,2,-3]
Output: 3
Explanation: Subarray [3] and [3,-2,2] both have maximum sum 3
```

Example 5:

```
Input: [-2,-3,-1]
Output: -1
Explanation: Subarray [-1] has maximum sum -1
```

Note:

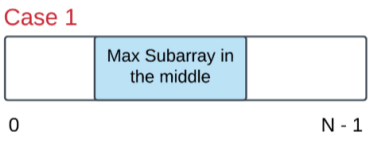
1. `-30000 <= A[i] <= 30000`
2. `1 <= A.length <= 30000`

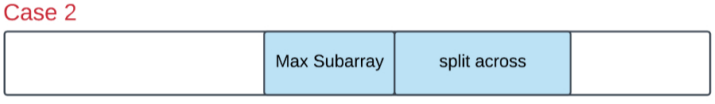
Hide Hint #1

For those of you who are familiar with the **Kadane's algorithm**, think in terms of that. For the newbies, Kadane's algorithm is used to finding the maximum sum subarray from a given array. This problem is a twist on that idea and it is advisable to read up on that algorithm first before starting this problem. Unless you already have a great algorithm brewing up in your mind in which case, go right ahead!

Hide Hint #2

What is an alternate way of representing a circular array so that it appears to be a straight array? Essentially, there are two cases of this problem that we need to take care of. Let's look at the figure below to understand those two cases:





0	N - 1	2N - 1

Hide Hint #3

The first case can be handled by the good old Kadane's algorithm. However, is there a smarter way of going about handling the second case as well?

Java

```java
class Solution {
    public int maxSubarraySumCircular(int[] A) {
        int sum=0;
        for(int i=0;i<A.length;i++){
            sum+=A[i];
        }
        int maxSubArray=kadane(A,0);
        int minSubArray=kadane(A,1);
        if((sum-minSubArray)!=0 && (sum-minSubArray)>=maxSubArray)
            return sum-minSubArray;
        else
            return maxSubArray;
    }
    public int kadane(int[] A,int sign){
        if(sign==0){
           int max=Integer.MIN_VALUE;
           int[] dp=new int[A.length];
           dp[0]=A[0];
           for(int i=1;i<dp.length;i++){
               dp[i]=Math.max(dp[i-1]+A[i],A[i]);
               max=Math.max(max,dp[i]);
           }
            return max==Integer.MIN_VALUE?dp[0]:max;
        }
        else{
           int min=Integer.MAX_VALUE;
           int[] dp=new int[A.length];
           dp[0]=A[0];
           for(int i=1;i<dp.length;i++){
               dp[i]=Math.min(dp[i-1]+A[i],A[i]);
               min=Math.min(min,dp[i]);
           }
            return min==Integer.MAX_VALUE?dp[0]:min;
        }
    }
}
```

Custom Testcase (Contribute)

Run Code Submit

[5,3,5]

How to create a testcase

Submission Result: Accepted More Details

Share your acceptance!

10