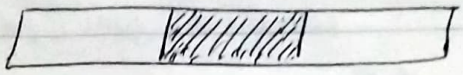
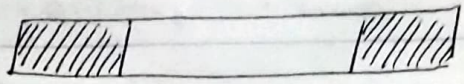
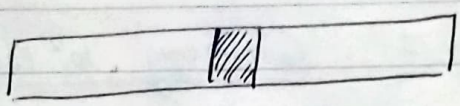
918: Maximum Sum Circular Subarray

Cases

All examples of leetcode show the above 3 cases:

1)  → Non-circular

2)  → circular

3)  → If all numbers are negative then smallest 'abs(num)' negative number.

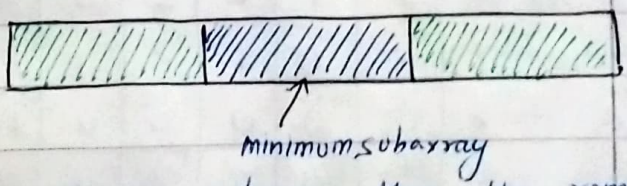
(3) case -

(1) case :- maximum is within the array (without circular iteration)

(2) case :- circular iteration gives the maximum subarray [5, -3, 5]

Algo

Intuition



If we calculate the minimum subarray then the remaining elements would be the maximum subarray.

1) Calculate maximum subarray using Kadane's Algorithm

2) Calculate minimum subarray using Kadane's Algorithm.
(Both at the same iteration.

```
for num in nums
    curr_max = max(curr_max + num, num)
    total_max = max(total_max, curr_max)
```

3) Return the maximum of (1) and [total - (2)]
max(total_max, total - total_min)

4) In case all elements are -ve return minimum abs(num)

-5, -3, -2, -6 → Return [-2]