Max Sum Subarray

Algorithm

```
curr_sum = 0
max_so_far = a[0]
start = 0
end = 0
temp = 0
for i in range(0, size):
    curr_sum = curr_sum + arr[i]
    if (max_so_far < curr_sum):
        start = temp
        end = i
        max_so_far = curr_sum
    if (curr_sum < 0):
        curr_sum = 0
        temp = i+1
```

Example:

	0	1	2	3	4	5	6	7	8	9	10	11	12
arr =	4	-3	-2	2	3	1	-2	-3	6	-6	-4	2	1

Start = 0 end = 0 temp = 0 i = 0

Step 1:

curr_sum = 0 + 4 = 4 [Both if conditions fail]

Step 2:

Start = 0 end = 0 curr_sum = 4 - 3 = 1 [BICF] i = 1

Step 3:

i = 2 Start = 0 end = 0 curr_sum = 1 - 2 = -1 ⇒ curr_sum = 0
temp = 3

Step_4:

i=3 curr_sum = 0+2 = 2

max_of_far < curr_sum ⇒ 4<2 ⇒ False

curr_sum < 0 ⇒ False

Step_5:

i=4 curr_sum = 2+3 = 5

max_so_far < curr_sum ⇒ 4<5 ⇒ True

↳ start = temp = 3 end = i = 3 max_so_far = 5

Step_6:

i=5 curr_sum = 5+1 = 6

max_so_far < curr_sum ⇒ 5<6 ⇒ True

↳ start = temp = 3 end = i = 5 max_so_far = 6

Step_7:

i=6 curr_sum = 6-2 = 4 [BICF]

Step_8:

i=7 curr_sum = 4-3 = 1 [BICF]

Step_9:

i=8 curr_sum = 1+6 = 7

max_so_far < curr_sum ⇒ 6<7 ⇒ True

↳ start = temp = 3 end = 8 = i max_so_far = 7

Step_10:

i=9 curr_sum = 7-6 = 1 [BICF]

Step_11:

i=10 curr_sum = 1-4 = -3

↳ curr_sum = 0 & temp = 10+1 = 11

max_so_far = 7 start = 3 end = 8

Step 12:

i=11 curr_sum = 0+2=2 [BICI]

Step 13:

i=12 curr_sum = 2+1 [BICF]

Result:

maximum sum of subarray = 7

Subarray indexes [3, 8]