Date: Day:

HomeWork 4

Design and Analysis of Algorithm

Name: Muhammad Hamza Khan
Roll: 21L-5654

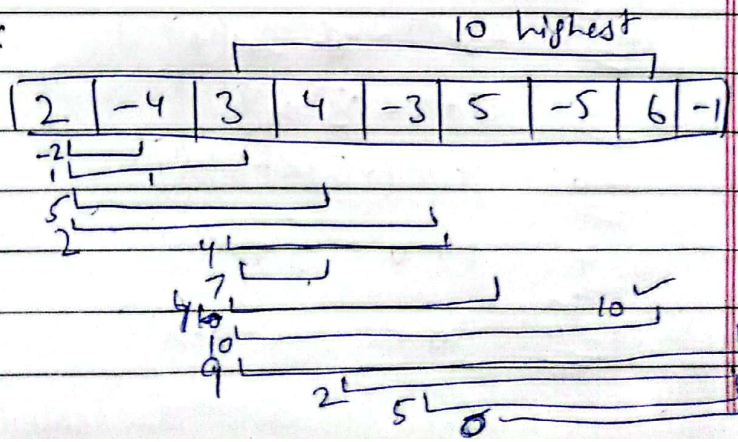
Task 1:

We can track the ~~end~~ start of subarray by intializing a global start vaBiable and updating it when our A[i] gets greater than MaxSum[i-1] + A[i].

Code:

```
MaxSubArraySum (A,n) {
    globalsum = MaxSum[1] = A[1]
    for ( i = 2 — n) {
        if (MaxSum[i-1] + A[i] > A[i])
            MaxSum[i] = MaxSum[i-1] + A[i]
        else
            { MaxSum[i] = A[i]
            ⟹ globalstart = i }
        if (globalsum < MaxSum[i])
            globalSum = MaxSum[i];
            globalEnd = i
    } return globalsum
}
```

Task 2:



Date: Day:

We iterate the whole array in $O(n^2)$.

globalsum = ~~2~~ ~~8~~ ~~7~~ ~~9~~ 10

MaxSum	2	-2	3	7	4	9	4	10	9

i = 1: A[1] = 2
i = 2: A[2] = -2
i = 3: A[3] = 3
i = 4: A[4] = 7
i = 5: A[5] = 4
i = 6: A[6] = 9
i = 7: A[7] = 4
i = 8: A[8] = 10
i = 9: A[9] = 9

Task3:

Dry Run Kadane's algo with MaxSum array

globalsum = ~~2~~ ~~8~~ ~~7~~ ~~9~~ 10

	0	1	2	3	4	5	6	7	8
MaxSum	2	-2	3	7	4	9	4	10	9
A	2	-4	3	4	-3	5	-5	6	-1

i = ~~1~~: $-4+2 < 2$
$-2 < 2$ ✗ ⇒ (2)

i = ~~2~~: $-2+3 < -2$
⇒ (-2)

i = ~~3~~: $3+4 < 3$ ⇒ (7)

i = ~~4~~: $-3+7 < -3$ ⇒ (4)

i = ~~5~~: $5+4 < 5$ ⇒ (9)

i = 6: $9+(-5) < -5$ ⇒ (9)

i = ~~7~~: $4+6 < 6$ ⇒ (10)

i = ~~8~~: $10+(-1) < -1$ ⇒ (9)

Task 4: Yes

Yes. For solving this problem in O(1) memory. We just have to use MaxSum as an integer instedd of array, pseudocode is as follows:

```
MaxSubArraySum(int n){
    int globalSum = A[0] = MaxSum
    int MaxSum = A[0]
    // array A starts with 0 index.
    for( i = 1 ; i<n ; i++ ){
        if(MaxSum + A[i] > A[i])
            MaxSum = MaxSum + A[i];
        else
            MaxSum = A[i];
        if(globalSum < MaxSum)
            globalSum = MaxSum;
    }
    return globalSum;
}
```