

Case - 1

```
In [2]: a = []  
max_subarray_sum(a,len(a))  
  
Array is empty
```

Case - 2

```
In [3]: a = [1]  
max_subarray_sum(a,len(a))  
  
1  
Maximum contiguous sum is 1
```

Case - 3

```
In [4]: a = [1, 2, 3, 4]  
max_subarray_sum(a,len(a))  
  
1  
2  
3  
4  
Maximum contiguous sum is 10
```

Case - 4

```
In [5]: a = [-7,-4,-2,-8]  
max_subarray_sum(a,len(a))  
  
-2  
Maximum contiguous sum is -2
```

Case - 5

```
In [6]: a = [-2, 3, 5, -7]  
max_subarray_sum(a,len(a))  
  
3  
5  
Maximum contiguous sum is 8
```

Case - 6

```
In [7]: a = [-2, -3, 4, -1, -2, 1, 5, -3]  
max_subarray_sum(a,len(a))  
  
4  
-1  
-2  
1  
5  
Maximum contiguous sum is 7
```

Case - 7

```
In [8]: a = [-2, 1, -3, 4, -1, 2, 1, -5, 4]  
max_subarray_sum(a,len(a))  
  
4  
-1  
2  
1  
Maximum contiguous sum is 6
```

Time Complexity: $O(n)$

Only one loop executing n times.