1. Maximum Sum subarray

```
Find the contiguous subarray within an array (containing at least one number) which
has the largest sum.

For example, given the array [2,3,-2,-3, 4],
the contiguous subarray [2,3] has the largest sum = 5.
```

Solution:

We need keep 2 Sum values, one is for current sum, one is for the max sum in the global loop.

For the i'th item a[i] comes, the current sum, denoted as curSum[i], is defined as the max subarray sum value ends with a[i]:

curSum[i] = max{curSum[i-1]+a[i], a[i]}

And the max sum, denoted as maxSum, is defined as

maxSum = max{curSum[i], maxSum}

For examples, array= {1, -2, 3, 10, -4, 7, 2, -5}, then

curSum ： 1 -1 3 13 9 16 18 13
maxSum ： 1 1 3 13 13 16 18 18

2. Maximum Product Subarray

```
Find the contiguous subarray within an array (containing at least one number) which
has the largest product.

For example, given the array [2, 3, 0.5, -2, 4],
the contiguous subarray [2,3] has the largest product = 6.
```

Solution:

If the elements in the array are positive, it is same as the max sum subarray:

For the i'th item a[i] comes, the current product, denoted as curProd [i], is defined as the max subarray product value ends with a[i]:

curProd[i] = max{curProd[i-1]*a[i], a[i]}

And the max product, denoted as maxProd, is defined as

maxProd = max{curProd [i], maxProd}

If the elements in the array are positive or negative, it is more complex than single polarity.

For the i'th item a[i] comes, max product ends with a[i] is determined by a[i] and max product/min product for item i-1. For example, a[i] = -4. Max product of item i-1 = 6, min product of item i-1 = -5. Then the max product of item i is -5 * (-4) = 20. If a[i] =4, the max is 4 * 6 = 24.

So we keep two states at item i-1: the max product and min product. If a[i] = 4, max product of item i-1 is 0.25, then max product at item i is 4.

Hence, we obtain the transfer function. Denote maxCurProd[i] as the max subarray product value ends with a[i], minCurProd[i] as the min subarray product value ends with a[i].

To calculate max product:

temp = max{maxCurProd[i-1]*a[i], minCurProd[i-1]*a[i]}

maxCurProd[i] = max{a[i], temp}

To calculate min product:

temp = min{maxCurProd[i-1]*a[i], minCurProd[i-1]*a[i]}

minCurProd[i] = min{a[i], temp}

For examples, array= {1, -2, 3, 10, -4}, then

maxCurProd：	1	-2	3	30	240
minCurProd：	1	-2	-6	-60	-120

Note the subarray with max product ending with a[i] can be different from the subarray with min product ending with a[i].