053. Maximum Subarray

053 Maximum Subarray

- **Divide and Conquer**+array
- **Dynamic Programming**+array

Description

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

For example, given the array `[-2,1,-3,4,-1,2,1,-5,4]`,
the contiguous subarray `[4,-1,2,1]` has the largest sum = `6`.

click to show more practice.

Seen this question in a real interview before? Yes No

1. Thought line

2. Divide and Conquer+array

```
class Solution {
private:
    int divideAndConquerMaxSubArray(vector<int>& nums, int st, int ed){
        if (st>ed) return INT_MIN;
        if (st==ed) return nums[st];
        int mid = (st+ed)/2;
        int lf_maxSubArraySum = divideAndConquerMaxSubArray(nums, st, mid-1);
        int rt_maxSubArraySum = divideAndConquerMaxSubArray(nums, mid+1, ed);

        // Calculate the possible result crossing middle point.
        int midToLeft = INT_MIN, midToright = INT_MIN;
        for (int i = mid, sum=0; i>=st; --i){
            sum+=nums[i];
            midToLeft = midToLeft>sum?midToLeft:sum;
        }
        for (int i = mid, sum=0; i<=ed; ++i){
            sum+=nums[i];
            midToright = midToright>sum?midToright:sum;
        }
        int mid_maxSubArraySum = midToLeft+midToright-nums[mid];
        return max(lf_maxSubArraySum,max(rt_maxSubArraySum,mid_maxSubArraySum));
    }

public:
    int maxSubArray(vector<int>& nums) {
        return divideAndConquerMaxSubArray(nums, 0, nums.size()-1);
    }
};
```