

## 53. Maximum Subarray

---

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return *its sum*.

A subarray is a contiguous part of an array.

Example 1:

```
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: [4,-1,2,1] has the largest sum = 6.
```

Example 2:

```
Input: nums = [1]
Output: 1
```

Example 3:

```
Input: nums = [5,4,-1,7,8]
Output: 23
```

Constraints:

- `1 <= nums.length <= 3 * 104`
- `-105 <= nums[i] <= 105`

- ```
class Solution:
    def maxSubArray(self, nums: List[int]) -> int:
        dp = [0]*len(nums)
        dp[0] = nums[0]
        for i in range(1, len(nums)):
            dp[i] = max(nums[i], dp[i-1]+nums[i])
        return max(dp)
```