Problem 53: Maximum Subarray

Problem Information

Difficulty: Medium
Acceptance Rate: 52.65%
Paid Only: No
Tags: Array, Divide and Conquer, Dynamic Programming

Problem Description

Given an integer array `nums`, find the subarray with the largest sum, and return _its sum_.

Example 1:

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

Example 2:

Input: nums = [1] **Output:** 1 **Explanation:** The subarray [1] has the largest sum 1.

Example 3:

Input: nums = [5,4,-1,7,8] **Output:** 23 **Explanation:** The subarray [5,4,-1,7,8] has the largest sum 23.

Constraints:

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

Follow up: If you have figured out the `O(n)` solution, try coding another solution using the **divide and conquer** approach, which is more subtle.

Code Snippets

C++:

```
class Solution {
public:
int maxSubArray(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int maxSubArray(int[] nums) {

}
}
```

Python3:

```
class Solution:
def maxSubArray(self, nums: List[int]) -> int:
```