Documentation for maxProduct Function

Overview

The maxProduct function is designed to find the maximum product of any contiguous subarray within a given integer array. It uses a dynamic programming approach to efficiently compute the maximum product while considering the effects of negative numbers on the product values.

Function Signature

```
def maxProduct(self, nums: List[int]) -> int:
```

Parameters

- nums (List[int]): A list of integers where each integer falls within the range \([-10, 10]\). The length of this list is guaranteed to be between 1 and 20,000 inclusive.

Returns

- int: The maximum product of any contiguous subarray within the nums list.

Description

The function computes the maximum product of contiguous subarrays using the following approach:

1. **Initialization:**

- *The function starts by initializing three variables:*
  - ***max_product:*** Tracks the maximum product found up to the current index in the array.
  - ***min_product:*** Tracks the minimum product found up to the current index in the array. This is necessary because a negative number can flip the sign of the product, making a large negative product potentially become large positive when multiplied by another negative number.
  - ***result:*** Holds the overall maximum product found so far.

2. **Iteration:**

- The function iterates through the nums list, starting from the second element.
- ***For each element:***
  - If the element is negative, it swaps max_product and min_product. This step is crucial because a negative number can turn a large negative product into a potentially large positive product.
  - It updates max_product to be the maximum of the current element alone or the product of the current element and the previous max_product.
  - It updates min_product to be the minimum of the current element alone or the product of the current element and the previous min_product.
  - It updates result to be the maximum of the current result and the updated max_product.

3. **Return:**

- After processing all elements, the function returns result, which holds the maximum product of any contiguous subarray found.

Constraints

- The function assumes that the input list nums contains at least one element.
- The function guarantees that the final product will fit within a 32-bit integer.

Edge Cases

- **Single Element:** If the list contains only one element, the function will return that element as the maximum product.

- **All Negative Numbers:** The function correctly handles cases where the entire list consists of negative numbers by appropriately swapping and updating max_product and min_product.

- **Zeroes:** The function properly handles zeroes by resetting the products when encountering zero, since multiplying by zero will result in zero.

Example

For the input list nums = [2, 3, -2, 4]:

- The function will iterate through the list and compute the maximum product subarray, which is [2, 3] with a product of 6.

For the input list nums = [-2, 0, -1]:

- The function will correctly identify that the maximum product subarray is [0] with a product of 0, since including negative numbers would not yield a larger product in this case.