

26. You are given an integer array `nums` with no duplicates. A maximum binary tree can be built recursively from `nums` using the following algorithm: Create a root node whose value is the maximum value in `nums`. Recursively build the left subtree on the subarray prefix to the left of the maximum value. Recursively build the right subtree on the subarray suffix to the right of the maximum value. Return the maximum binary tree built from `nums`.

Code:

```
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def construct_maximum_binary_tree(nums):
    if not nums:
        return None

    max_index = nums.index(max(nums))

    root = TreeNode(nums[max_index])

    if max_index != len(nums) - 1:
        root.left = construct_maximum_binary_tree(nums[:max_index])
        root.right = construct_maximum_binary_tree(nums[max_index + 1:])

    return root

nums = [3, 2, 1, 6, 0, 5]
root = construct_maximum_binary_tree(nums)
```

Output:

```
<__main__.TreeNode object at 0x000001B0C7F14680>
```

Time Complexity:

- $T(n) = O(n)$