Previous Greater Element of Array Element

Given an array of integers, for each element, find the previous element in the array that is greater than the current element. If no such element exists, assign -1 to that position in the result array.

Example:

Input: [2, 4, 9, 3, 1]

Output: [-1, -1, -1, 9, 3]

Approach 1: Function to find the previous greater elements using a brute force approach

In this approach, for each element in the array, we iterate through the previous elements to find the first element that is greater than the current element. If found, we update the result array with the greater element; otherwise, we assign -1.

Time Complexity: This approach has a time complexity of O(n^2) where n is the size of the array, as we are using nested loops.

Space Complexity: The space complexity is O(n) for the result array.

Approach 2: Function to find the previous greater elements using a stack-based approach

In the stack-based approach, we use a stack to keep track of potential previous greater elements. We iterate through the array in forward order. For each element, we pop elements from the stack until finding an element greater than the current element. We update the result array accordingly.

Time Complexity: This approach has a time complexity of O(n), where n is the size of the array, as each element is pushed and popped from the stack only once.

Space Complexity: The space complexity is O(n) as well, for the stack and the result array.