<https://www.geeksforgeeks.org/write-a-c-function-to-print-the-middle-of-the-linked-list/>

<https://leetcode.com/problems/next-greater-node-in-linked-list/>

1019. Next Greater Node In Linked List

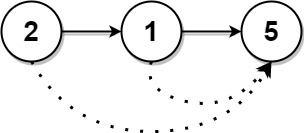
Medium

You are given the head of a linked list with n nodes.

For each node in the list, find the value of the **next greater node**. That is, for each node, find the value of the first node that is next to it and has a **strictly larger** value than it.

Return an integer array answer where answer[i] is the value of the next greater node of the ith node (**1-indexed**). If the ith node does not have a next greater node, set answer[i] = 0.

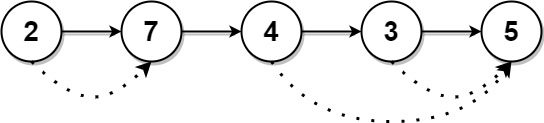
**Example 1:**



**Input:** head = [2,1,5]

**Output:** [5,5,0]

**Example 2:**



**Input:** head = [2,7,4,3,5]

**Output:** [7,0,5,5,0]

**Constraints:**

* The number of nodes in the list is n.
* 1 <= n <= 104
* 1 <= Node.val <= 109
* Very similar to this problem [503. Next Greater Element II](https://leetcode.com/problems/next-greater-element-ii/discuss/98270/)

## ****Time Complexity****:

* O(N) Time, O(N) Space  
    
    
  **Java:**  
  Transform the linked list to an arraylist,  
  then it's a normal "next larger element" problem,  
  solved by stack.
* public int[] nextLargerNodes(ListNode head) {
* ArrayList<Integer> A = new ArrayList<>();
* for (ListNode node = head; node != null; node = node.next)
* A.add(node.val);
* int[] res = new int[A.size()];
* Stack<Integer> stack = new Stack<>();
* for (int i = 0; i < A.size(); ++i) {
* while (!stack.isEmpty() && A.get(stack.peek()) < A.get(i))
* res[stack.pop()] = A.get(i);
* stack.push(i);
* }
* return res;
* }