16TH JAN 2023

FIRST :-

Linked List that is Sorted Alternatingly :: Easy

Given a Linked list of size **N**, the list is in alternating ascending and descending orders. Sort the given linked list in non-decreasing order.

Example 1:

```
Input:
LinkedList: 1->9->2->8->3->7
Output: 1 2 3 7 8 9
Explanation: After sorting the given
list will be 1-> 2-> 3-> 7-> 8-> 9.
```

Example 2:

```
Input:
LinkedList: 13->99->21->80->50
Output: 13 21 50 80 99
Explanation: After sorting the given list
will be 12-> 21-> 50-> 80-> 99.
```

Your Task:

You do not need to read input or print anything. The task is to complete the function **sort**() which should sort the linked list in non-decreasing order.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

1 <= Number of nodes <= 100

0 <= Values of the linked list <= 10^3

CODE SECTION:-

```
void sort(Node **head)
{
    // Code here
    // brute force method
    vector<int> v;
    Node *p = (*head);

    while (p)
    {

        v.push_back(p->data);
        p = p->next;
    }

    sort(v.begin(), v.end());
    p = (*head);
    int i = 0;
    while (p)
    {
        p->data = v[i];
        i++;
        p = p->next;
    }
}
```

-: Done for the today :-