https://course.acciojob.com/idle?question=a0507fee-f1ea-4df2-a4d0-cbbfccf096c4

- MEDIUM
- Max Score: 40 Points

Merge two sorted linked list

Merge two sorted linked lists and return it as a sorted list. The list should be made by splicing together the nodes of the first two lists.

Input Format

The format for each test case is as follows:

The first line contains an integer `n`, the length of the first linked list.

The next line contain `n` integers, the elements of the linked list.

The next line contains an integer `m`, the length of the second linked list.

The next lines contain `m` integers, the elements of the second linked list.

Output Format

Output a single line of `(n + m)` integers consisting all elements of linked lists in sorted order.

Example 1

Input

```
3
1 2 4
3
1 3 4
```

Output

```
1 1 2 3 4 4
```

Explanation

Merge the two linked list

Example 2

Input

```
3
1 5 9
3
1 3 4
```

Output

```
1 1 3 4 5 9
```

Explanation

Merge the two linked list in sorted order

Constraints

The number of nodes in both lists is in the range `[0, 50]`.

`-100 <= Node.val <= 100`

Both `list1` and `list2` are sorted in non-decreasing order.

Topic Tags

- Sorting
- Linked lists

My code

```
// n java
import java.util.*;
import java.lang.*;
import java.io.*;

class Node
{
    int data;
    Node next;

    Node(int data, Node next)
    {
        this.data = data;
        this.next = next;
    }

    Node() {}
}


public class Main
{
static void display(Node h)
  {
    Node p=h;
    while(p!=null)
      {
```

```
        System.out.print(p.data+" ");
        p=p.next;
      }
  }

static Node sortedMerge(Node a, Node b)
    {
        // a dummy first node to hang the result on
        Node dummy = new Node();

        // points to the last result node — so `tail.next` is the place
        // to add new nodes to the result.
        Node tail = dummy;

        while (true)
        {
            // if either list runs out, use the other list
            if (a == null)
            {
                tail.next = b;
                break;
            }
            else if (b == null)
            {
                tail.next = a;
                break;
            }

            if (a.data <= b.data)
            {
```

```
            if (a != null)
            {
                Node newNode = a;
                a = a.next;

                newNode.next = tail.next;
                tail.next = newNode;
            }
        }
        else {
            if (b != null)
            {
                Node newNode = b;
                b = b.next;

                newNode.next = tail.next;
                tail.next = newNode;
            }
        }
        tail = tail.next;
    }

    return dummy.next;
  }


    public static void main (String[] args) throws
java.lang.Exception
    {
        //your code here
```

```
Scanner s=new Scanner(System.in);
      Node a=null,b=null;

      int n=s.nextInt();    int arr[]=new int[n];
for(int i=0;i<n;i++)
  arr[i]=s.nextInt();

  int m=s.nextInt();   int arrb[]=new int[m];
      for(int i=0;i<m;i++)
     arrb[i]=s.nextInt();
for(int i=n;i>0;i--)
   a=new Node(arr[i-1], a);
for(int i=m;i>0;i--)
   b=new Node(arrb[i-1], b);


      Node d=sortedMerge(a,b);
      display(d);
```

```
        }
    }
```