



Dash



All



Articles



Videos



Problems



Quiz



Contest

<< Prev

Next >>

Merge two sorted linked lists

Given two sorted linked lists consisting of **N** and **M** nodes respectively. The task is to merge both of the lists (in place) and return the head of the merged list.

Examples:

Input: *a: 5->10->15, b: 2->3->20*

Output: *2->3->5->10->15->20*

Input: *a: 1->1, b: 2->4*

Output: *1->1->2->4*

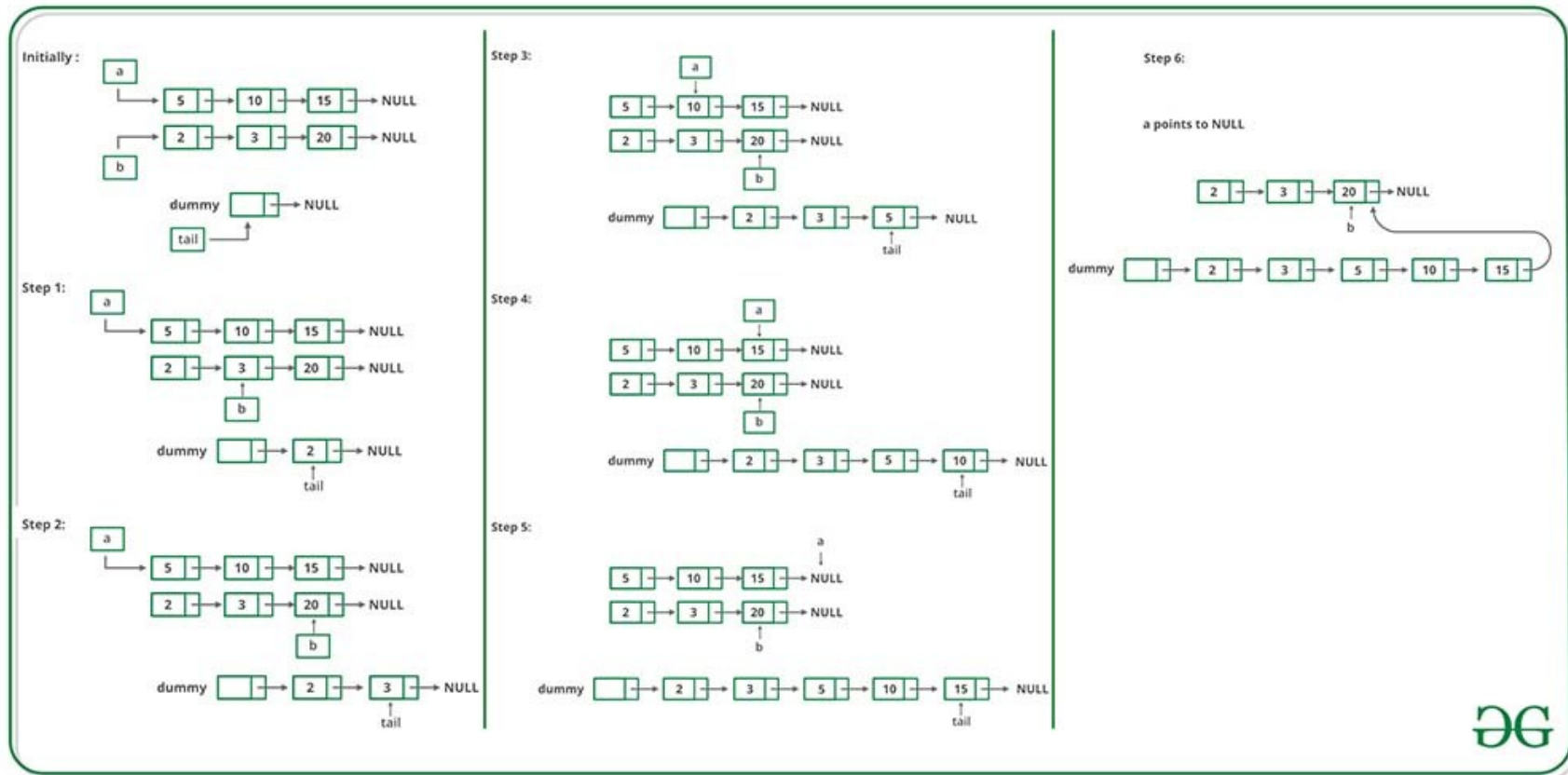
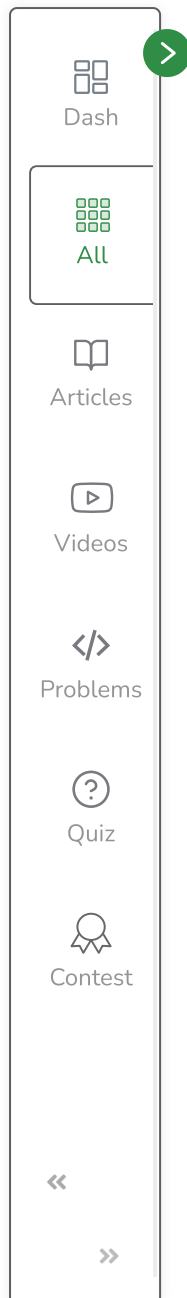
Merge two sorted linked lists using Dummy Nodes:

The idea is to use a temporary dummy node as the start of the result list. The pointer Tail always points to the last node in the result list, so appending new nodes is easy.

Follow the below illustration for a better understanding:

Illustration:





Follow the steps below to solve the problem:

- First, make a dummy node for the new merged linked list
- Now make two pointers, one will point to **list1** and another will point to **list2**.
- Now traverse the lists till one of them gets exhausted.
- If the value of the node pointing to either list is smaller than another pointer, add that node to our merged list and increment that pointer.

Below is the implementation of the above approach:

C++

Java



Dash



All



Articles



Videos



Problems



Quiz



Contest



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

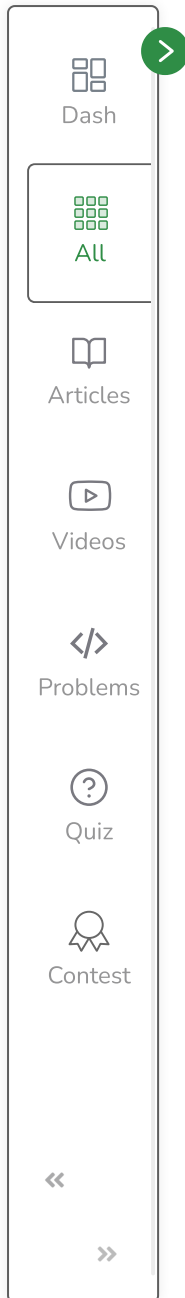
class Node {
    int data;
    Node next;
    Node(int x)
    {
        data = x;
        next = null;
    }
}

class Test {

    public static void main(String args[])
    {
        Node a = new Node(10);
        a.next = new Node(20);
        a.next.next = new Node(30);
        Node b = new Node(5);
        b.next = new Node(35);
        printlist(sortedMerge(a, b));
    }

    static Node sortedMerge(Node a, Node b)
    {
```





```
if (a == null)
    return b;
if (b == null)
    return a;
Node head = null, tail = null;
if (a.data <= b.data) {
    head = tail = a;
    a = a.next;
}
else {
    head = tail = b;
    b = b.next;
}
while (a != null && b != null) {
    if (a.data <= b.data) {
        tail.next = a;
        tail = a;
        a = a.next;
    }
    else {
        tail.next = b;
        tail = b;
        b = b.next;
    }
}
if (a == null) {
    tail.next = b;
}
else {
```





Dash



All



Articles



Videos



Problems



Quiz



Contest



```
        return head;
    }

    public static void printlist(Node head)
    {
        Node curr = head;
        while (curr != null) {
            System.out.print(curr.data + " ");
            curr = curr.next;
        }
        System.out.println();
    }
}
```

Output:

5 10 20 30 35

Time Complexity: $O(N + M)$, where N and M are the size of list1 and list2 respectively

Auxiliary Space: $O(1)$

Mark as Read

 Report An Issue

If you are facing any issue on this page. Please let us know.

