[LeetCode](https://leetcode.com/problems/merge-in-between-linked-lists/submissions/1208710536/?envType=daily-question&envId=2024-03-20)

<https://github.com/AdityaKonda6/-50DaysOfCoding>

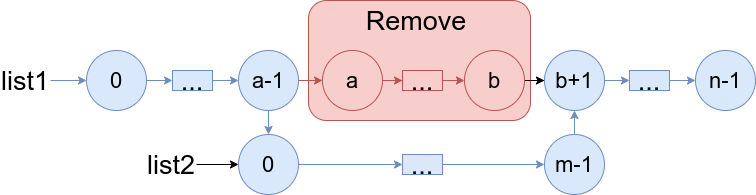
<https://leetcode.com/problems/merge-in-between-linked-lists/submissions/1208710536/?envType=daily-question&envId=2024-03-20>

<https://www.linkedin.com/in/aditya-adi-konda/>

Day 30 of [#50dayscodingchallenge](https://www.linkedin.com/feed/hashtag/?keywords=50dayscodingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633):  
[#leetcode](https://www.linkedin.com/feed/hashtag/?keywords=leetcode&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodechallenge](https://www.linkedin.com/feed/hashtag/?keywords=leetcodechallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodestreak](https://www.linkedin.com/feed/hashtag/?keywords=leetcodestreak&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode2024](https://www.linkedin.com/feed/hashtag/?keywords=leetcode2024&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode50day](https://www.linkedin.com/feed/hashtag/?keywords=leetcode50day&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
   
Ventured further into my coding journey today, tackling the engaging LeetCode Problem "Successfully solved LeetCode Problem 🎈💻“1669. Merge In Between Linked Lists.”  
   
✨ Task: You are given two linked lists: list1 and list2 of sizes n and m respectively.

Remove list1's nodes from the ath node to the bth node, and put list2 in their place.

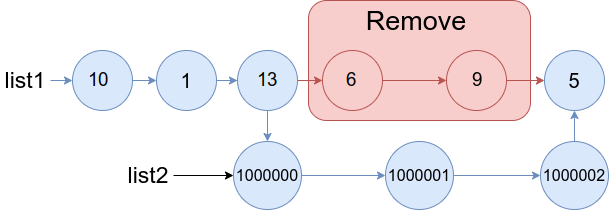
The blue edges and nodes in the following figure indicate the result:



Build the result list and return its head.

Examples:

Example 1:

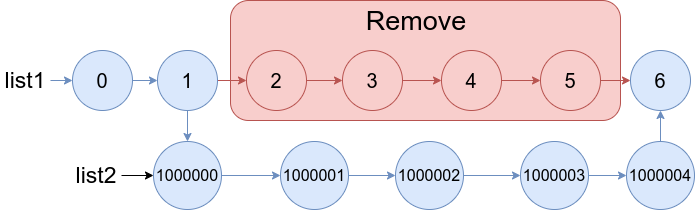


Input: list1 = [10,1,13,6,9,5], a = 3, b = 4, list2 = [1000000,1000001,1000002]

Output: [10,1,13,1000000,1000001,1000002,5]

Explanation: We remove the nodes 3 and 4 and put the entire list2 in their place. The blue edges and nodes in the above figure indicate the result.

Example 2:



Input: list1 = [0,1,2,3,4,5,6], a = 2, b = 5, list2 = [1000000,1000001,1000002,1000003,1000004]

Output: [0,1,1000000,1000001,1000002,1000003,1000004,6]

Explanation: The blue edges and nodes in the above figure indicate the result.

Let's Connect:

If you find this problem intriguing or have insights to share, let's connect! I'm passionate about problem-solving, algorithmic thinking, and collaborative learning. Feel free to comment or reach out for engaging discussions and knowledge exchange.Unravel the mystery using your coding skills!

[#CodingChallenge](https://www.linkedin.com/feed/hashtag/?keywords=codingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#Algorithm](https://www.linkedin.com/feed/hashtag/?keywords=algorithm&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#LinkedInPost](https://www.linkedin.com/feed/hashtag/?keywords=linkedinpost&highlightedUpdateUrns=urn:li:activity:7166316239483461633) #Algorithm #Optimization #DataStructures #CodingChallenge  
  
Excited about the progress and challenges ahead!  
   
Make Sure You Follow My GitHub For Solutions: <https://github.com/AdityaKonda6/-50DaysOfCoding>  
  
  
Happy coding!

**Solution:-**

class Solution {

  public ListNode mergeInBetween(ListNode list1, int a, int b, ListNode list2) {

    ListNode nodeBeforeA = list1;

    for (int i = 0; i < a - 1; ++i)

      nodeBeforeA = nodeBeforeA.next;

    ListNode nodeB = nodeBeforeA.next;

    for (int i = 0; i < b - a; ++i)

      nodeB = nodeB.next;

    nodeBeforeA.next = list2;

    ListNode lastNodeInList2 = list2;

    while (lastNodeInList2.next != null)

      lastNodeInList2 = lastNodeInList2.next;

    lastNodeInList2.next = nodeB.next;

    nodeB.next = null;

    return list1;

  }

}

