

Day-5



Problem-4: Remove Nth Node from
end of list.

Self

① → ② → ③ → ④ → ⑤
Head = [1, 2, 3, 0, 4, 5] n = 2, output: [1, 2, 3, 5]

⇒ Using while loop and count no. of node return node - n //

⇒ now delete node - n Node from list and print rest of the list.

TC: O(N)

SC: O(1)

Remarks

Date ___/___/_____

Page No. _____



Problem-5:

Add two number as linked list.

$d_1 = [2, 4, 3]$, $d_2 = [5, 6, 4]$
output: $[7, 0, 8]$

// Sum/add the value of both the list
data $d_1 \rightarrow$ data $+ d_2 \rightarrow$ data
if the sum is > 9 then it means it generates carry that we will add in another value.

// Logic: $sum + carry$
 $carry = sum / 10;$

TC: $O(\max(m, n))$

SC: $O(\max(1, \min(m, n)))$

Remarks _____

Problem-6: Delete Node in a linked list.



head = [4, 5, 1, 9], node = 5

// while traversing find node value
and skip the pointer and
move node \rightarrow next \rightarrow val.

// node \rightarrow val = node \rightarrow next \rightarrow val
node \rightarrow next = node \rightarrow next \rightarrow next
then return.

SC: $O(1)$
TC: $O(N)$

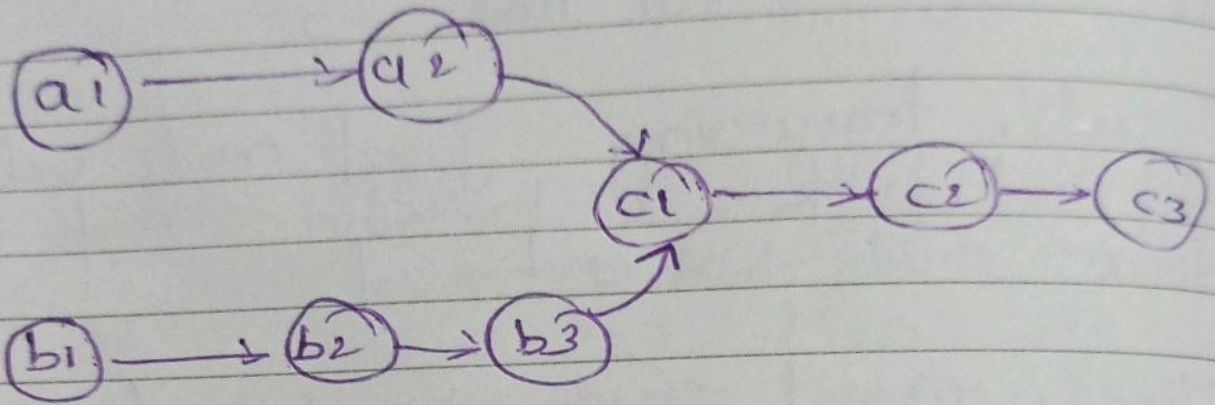
Remarks

ate __/__/____

Not



Problem-1: Intersection of two linked lists.



interval = 8, list A = [4, 1, 8, 7, 5]
list B = [5, 6, 1, 8, 4, 5], skip B = 3

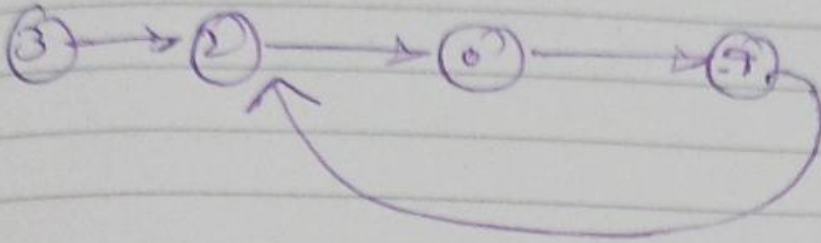
// If a and b list have different length then we will stop the loop after second iteration.

// for the end of first iteration, we just reset the pointers to the head of another ll.

a = a == NULL ? head B : a → next
b = b == NULL ? head A : b → next
return a;

Remarks

Problem-8: linked list cycle



head = [3, 2, 0, -4], pos = 1
output: true

// Use Floyd's method

// move fast pointer two step
slow pointer to one step

// where fast == slow

// return true (// cycle is present)

Remarks