5943. Delete the Middle Node of a Linked List

My Submissions (/contest/weekly-contest-270/problems/delete-the-middle-node-of-a-linked-list/submissions/)

Back to Contest (/contest/weekly-contest-270/)

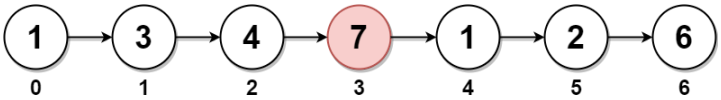
User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

You are given the `head` of a linked list. **Delete** the **middle node**, and return *the* `head` *of the modified linked list*.

The **middle node** of a linked list of size `n` is the $\lfloor n / 2 \rfloor^{th}$ node from the **start** using **0-based indexing**, where $\lfloor x \rfloor$ denotes the largest integer less than or equal to `x`.

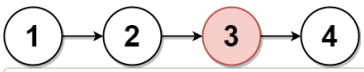
- For `n = 1, 2, 3, 4,` and `5`, the middle nodes are `0, 1, 1, 2,` and `2`, respectively.

Example 1:



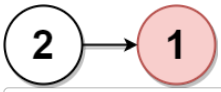

```
Input: head = [1,3,4,7,1,2,6]
Output: [1,3,4,1,2,6]
Explanation:
The above figure represents the given linked list. The indices of the nodes are written below.
Since n = 7, node 3 with value 7 is the middle node, which is marked in red.
We return the new list after removing this node.
```

Example 2:




```
Input: head = [1,2,3,4]
Output: [1,2,4]
Explanation:
The above figure represents the given linked list.
For n = 4, node 2 with value 3 is the middle node, which is marked in red.
```

Example 3:




```
Input: head = [2,1]
Output: [2]
Explanation:
The above figure represents the given linked list.
For n = 2, node 1 with value 1 is the middle node, which is marked in red.
Node 0 with value 2 is the only node remaining after removing node 1.
```

Constraints:

- The number of nodes in the list is in the range $[1, 10^5]$.
- $1 <= Node.val <= 10^5$

JavaScript

```
const getAllData = (list) => {
    let res = [];
    let current = list;
    while (current) {
        res.push(current.val);
```

```
        current = current.next;
    }
    return res;
};

const createL = (arr) => {
    let tmp, node = null;
    let n = arr.length;
    for (let i = n - 1; ~i; i--) {
        if (!node) {
            node = new ListNode(arr[i]);
        } else {
            tmp = new ListNode(arr[i]);
            tmp.next = node;
            node = tmp;
        }
    }
    return node;
};

const deleteMiddle = (head) => {
    let a = getAllData(head), n = a.length;
    let mid = n >> 1;
    a.splice(mid, 1);
    return createL(a);
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/597083988/) | More Details > (/submissions/detail/597083988/)

Share your acceptance!