75 Days of Code Day 30 328. Odd Even Linked List (leetcode)
Type : Linked List

Medium

Given the head of a singly linked list, group all the nodes with odd indices together followed by the nodes with even indices, and return the reordered list.

The first node is considered odd, and the second node is even, and so on.

Note that the relative order inside both the even and odd groups should remain as it was in the input.

You must solve the problem in O(1) extra space complexity and O(n) time complexity.

Example 1:

Input: head = [1,2,3,4,5]
Output: [1,3,5,2,4]
Example 2:

Input: head = [2,1,3,5,6,4,7]
Output: [2,3,6,7,1,5,4]

Solution using array

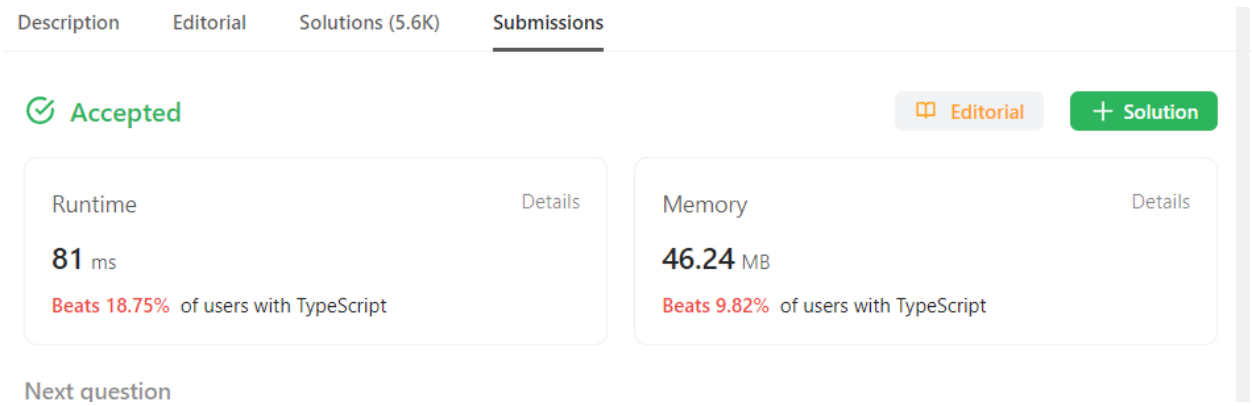
```
function oddEvenList(head: ListNode | null): ListNode | null {
  if (!head || !head.next) return head;

  let oddEvenArr: number[] = [];
  let temp = head;
  while (temp) {
    oddEvenArr.push(temp.val);
    temp = temp.next;
  }

  let oddElements = [];
  let evenElements = [];
  for (let index = 0; index < oddEvenArr.length; index++) {
    if (index % 2 === 1) {
      oddElements.push(oddEvenArr[index]);
    } else {
      evenElements.push(oddEvenArr[index]);
    }
  }

  let joinElementsInOrder = evenElements.concat(oddElements);

  let newListHead = new ListNode(joinElementsInOrder[0]);
  let current = newListHead;
  for (let index = 1; index < joinElementsInOrder.length; index++) {
    current.next = new ListNode(joinElementsInOrder[index]);
    current = current.next;
  }
  return newListHead;
}
```



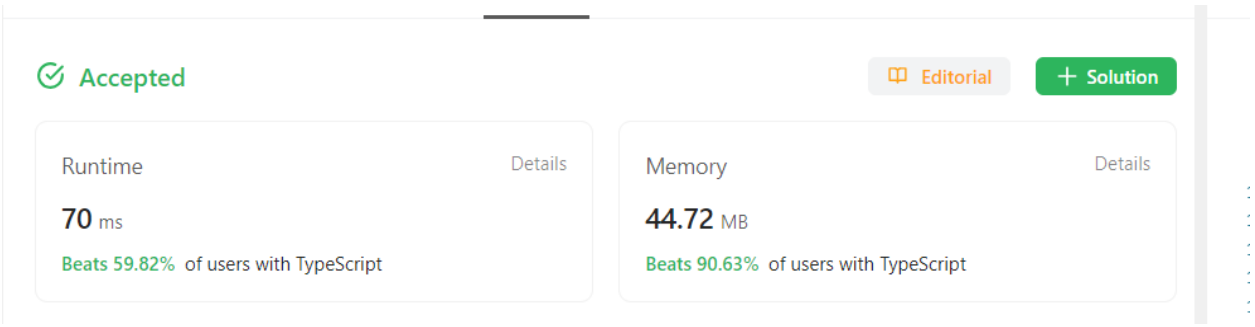
This solution fails o(1) space complexity so now

Shubham Agrahari

Solution using fast and slow algorithm

1. We are going to create two linked list one is even list and other is odd list and after that we link oddlist with even list

```
65
66  function oddEvenListOptimize(head: ListNode | null): ListNode | null {
67    if (!head || !head.next) return head;
68
69    let evenHead = head.next;
70    let oddHead = head;
71
72    let evenPointer = evenHead;
73    let oddPointer = oddHead;
74
75    while (evenPointer && evenPointer.next) {
76      oddPointer.next = evenPointer.next;
77      oddPointer = oddPointer.next;
78
79      evenPointer.next = oddPointer.next;
80      evenPointer = evenPointer.next;
81    }
82
83    oddPointer.next = evenHead;
84
85    return oddHead;
86  }
```



Shubham Agrahari