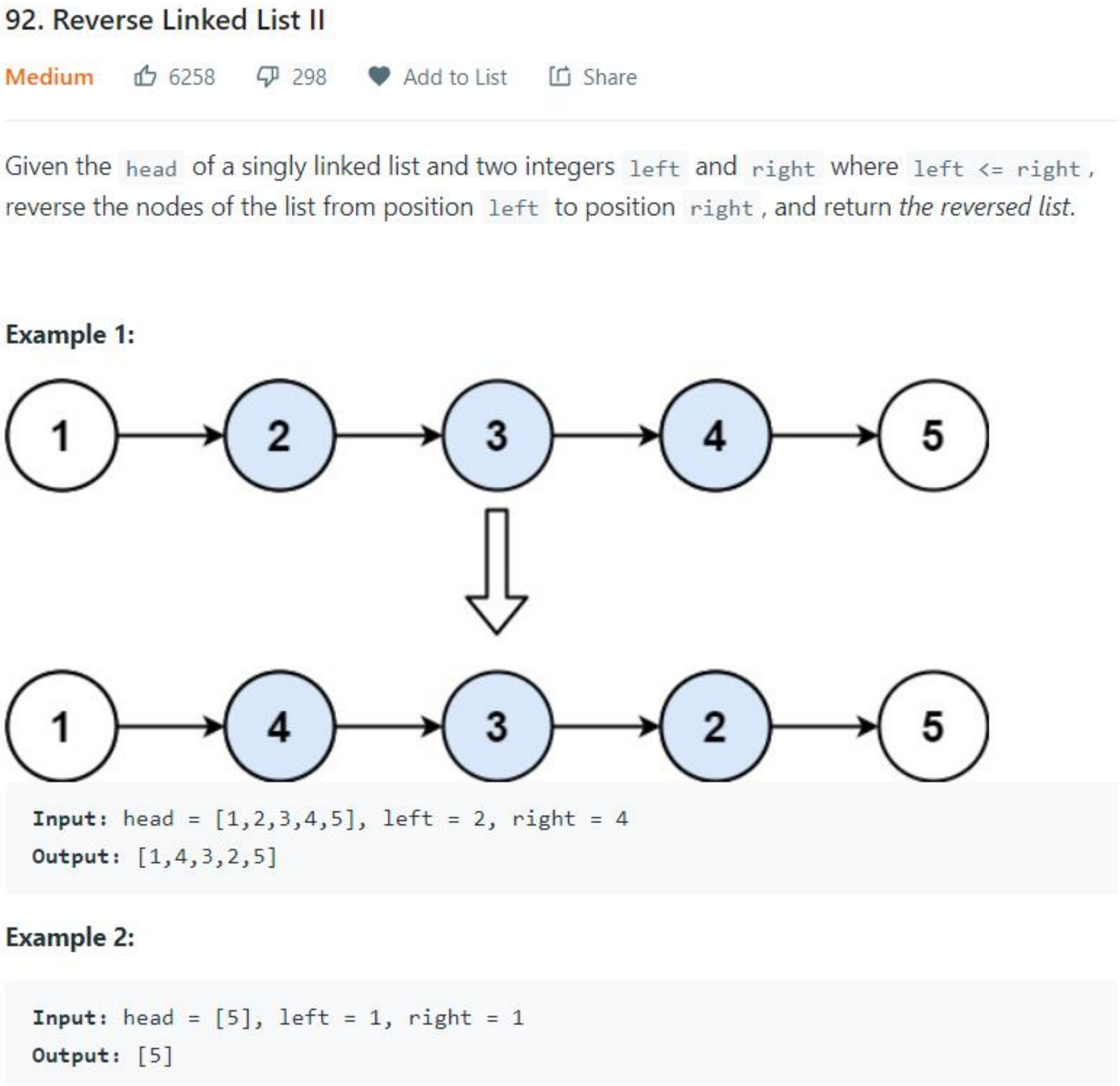


92. Reverse Linked List II

Medium 👍 6258 👎 298 ♥ Add to List Share

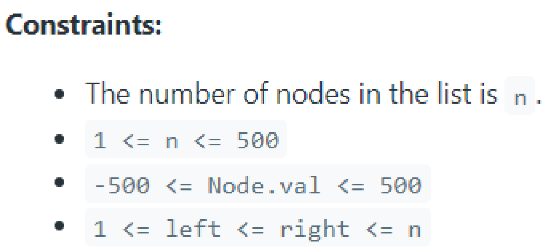
Given the `head` of a singly linked list and two integers `left` and `right` where `left <= right`, reverse the nodes of the list from position `left` to position `right`, and return *the reversed list*.

Example 1:

```
Input: head = [1,2,3,4,5], left = 2, right = 4
Output: [1,4,3,2,5]
```

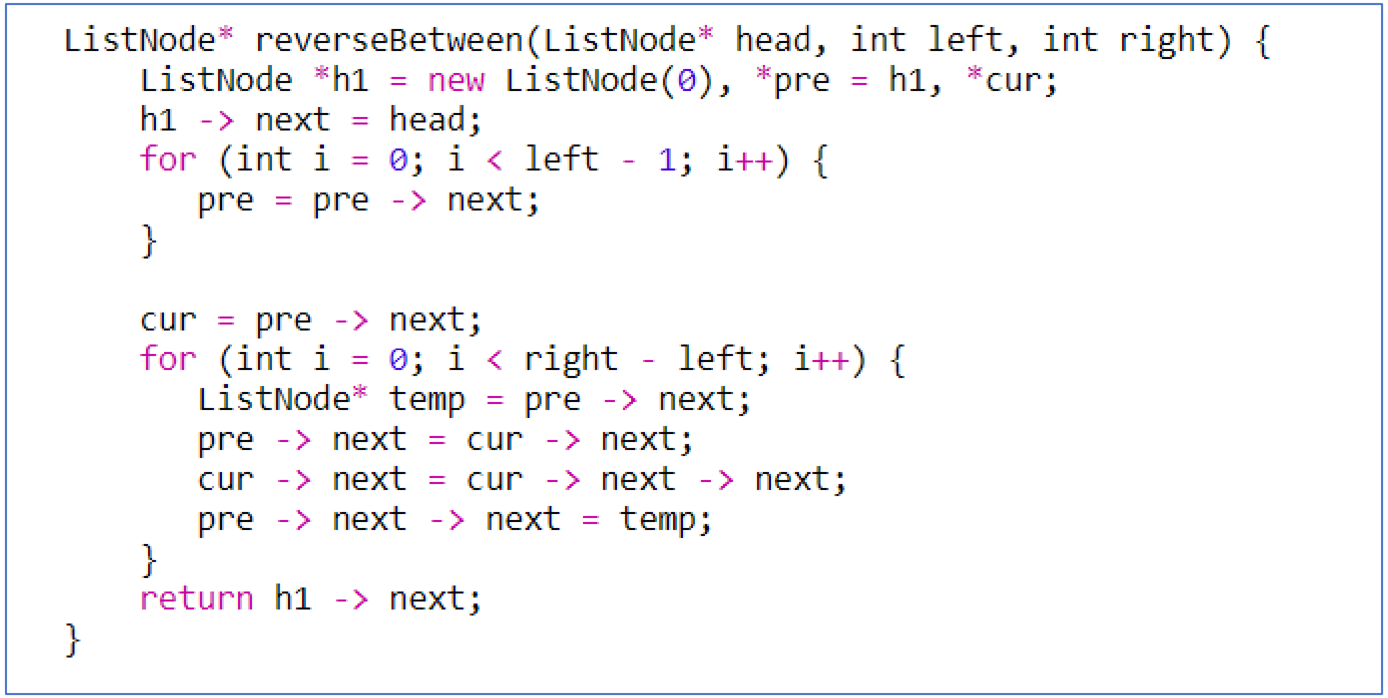
Example 2:

```
Input: head = [5], left = 1, right = 1
Output: [5]
```



Constraints:

- The number of nodes in the list is `n`.
- `1 <= n <= 500`
- `-500 <= Node.val <= 500`
- `1 <= left <= right <= n`



```
ListNode* reverseBetween(ListNode* head, int left, int right) {
    ListNode *h1 = new ListNode(0), *pre = h1, *cur;
    h1 -> next = head;
    for (int i = 0; i < left - 1; i++) {
       pre = pre -> next;
    }

    cur = pre -> next;
    for (int i = 0; i < right - left; i++) {
       ListNode* temp = pre -> next;
       pre -> next = cur -> next;
       cur -> next = cur -> next -> next;
       pre -> next -> next = temp;
    }
    return h1 -> next;
}
```

#100daysofDSA

/rvislive

Rakesh Vishwakarma