

## Grokking the Coding Interview: Patterns for Coding Questions

54% completed



Challenge 2

### Pattern: In-place Reversal of a LinkedList

- Introduction
- Reverse a LinkedList (easy)
- Reverse a Sub-list (medium)
- Reverse every K-element Sub-list (medium)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem Challenge 2**

### Pattern: Tree Breadth First Search

- Introduction
- Binary Tree Level Order Traversal (easy)
- Reverse Level Order Traversal (easy)
- Zigzag Traversal (medium)
- Level Averages in a Binary Tree (easy)
- Minimum Depth of a Binary Tree (easy)
- Level Order Successor (easy)
- Connect Level Order Siblings (medium)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem Challenge 2

### Pattern: Tree Depth First Search

- Introduction
- Binary Tree Path Sum (easy)
- All Paths for a Sum (medium)
- Sum of Path Numbers (medium)
- Path With Given Sequence (medium)
- Count Paths for a Sum (medium)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem Challenge 2

### Pattern: Two Heaps

- Introduction
- Find the Median of a Number Stream (medium)
- Sliding Window Median (hard)
- Maximize Capital (hard)
- Problem Challenge 1
- Solution Review: Problem

## Solution Review: Problem Challenge 2

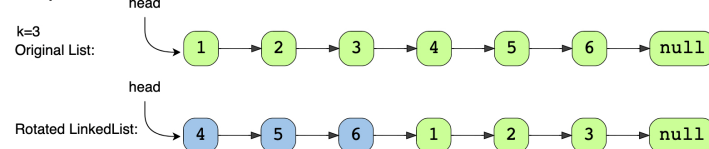
### We'll cover the following

- Rotate a LinkedList (medium)
- Solution
  - Code
  - Time complexity
  - Space complexity

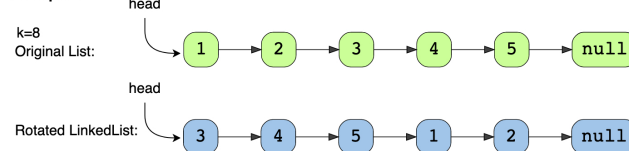
### Rotate a LinkedList (medium) #

Given the head of a Singly LinkedList and a number 'k', rotate the LinkedList to the right by 'k' nodes.

#### Example 1:



#### Example 2:



### Solution #

Another way of defining the rotation is to take the sub-list of 'k' ending nodes of the LinkedList and connect them to the beginning. Other than that we have to do three more things:

1. Connect the last node of the LinkedList to the head, because the list will have a different tail after the rotation.
2. The new head of the LinkedList will be the node at the beginning of the sublist.
3. The node right before the start of sub-list will be the new tail of the rotated LinkedList.

### Code #

Here is what our algorithm will look like:

Java Python3 C++ JS

```
1 class Node {
2   constructor(value, next = null) {
3     this.value = value;
4     this.next = next;
5   }
6
7   print_list() {
8     let temp = this;
9     while (temp !== null) {
10      process.stdout.write(`${temp.value} `);
11      temp = temp.next;
12    }
13    console.log();
14  }
15 }
16
17 function rotate(head, rotations) {
18   if (head === null || head.next === null || rotations <= 0) {
19     return head;
20   }
21
22   // find the length and the last node of the list
23   let last_node = head;
24   let list_length = 1;
25   while (last_node.next !== null) {
26     last_node = last_node.next;
27     list_length += 1;
28 }
```

RUN SAVE RESET

Close

Output 4 / 846s

Tracks

My Courses

Edpresso

Refer a Friend

Create

Challenge 1

Pattern: Subsets

Introduction

Subsets (easy)

Subsets With Duplicates (easy)

Permutations (medium)

String Permutations by changing case (medium)

Balanced Parentheses (hard)

Unique Generalized Abbreviations (hard)

Problem Challenge 1

Solution Review: Problem Challenge 1

Problem Challenge 2

Solution Review: Problem Challenge 2

Problem Challenge 3

Solution Review: Problem Challenge 3

Pattern: Modified Binary Search

Introduction

Order-agnostic Binary Search (easy)

Nodes of original LinkedList are: 1 2 3 4 5 6

Nodes of reversed LinkedList are: 4 5 6 1 2 3

Time complexity

The time complexity of our algorithm will be  $O(N)$  where 'N' is the total number of nodes in the LinkedList.

Space complexity

We only used constant space, therefore, the space complexity of our algorithm is  $O(1)$ .

Interviewing soon? We've partnered with Hired so that companies apply to you instead of you applying to them. [See how](#)

Back

Next

Problem Challenge 2

Introduction

Mark as Completed

Report an Issue

Ask a Question