Swapping Nodes in a Linked List

Try to solve the Swapping Nodes in a Linked List problem.

We'll cover the following

Statement

Given the linked list and an integer, `k`, return the head of the linked list after swapping the values of the kth node from the beginning and the kth node from the end of the linked list.

> **Note:** We'll number the nodes of the linked list starting from 1 to n.

Constraints:

- The linked list will have `n` number of nodes.
- $1 \leq$ `k` $\leq$ `n` ≤ 500
- $-5000 \leq$ `Node.value` ≤ 5000

Examples

Sample example 1

Input

`k = 1`

Output

1 of 5

Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Swapping Nodes in a Linked List

1

What is the output if the following data is given as input?

linked list = $4 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 0 \rightarrow 2$

k = 3

A) $8 \rightarrow 9 \rightarrow 0 \rightarrow 4 \rightarrow 2 \rightarrow 7 \rightarrow 2$

B) $7 \rightarrow 2 \rightarrow 4 \rightarrow 0 \rightarrow 9 \rightarrow 8 \rightarrow 2$

C) $4 \rightarrow 2 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 0 \rightarrow 2$

D) $2 \rightarrow 9 \rightarrow 0 \rightarrow 4 \rightarrow 2 \rightarrow 7 \rightarrow 8$

Submit Answer

Question 1 of 4
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Initialize a pointer `curr` with a head node and a variable `count` with 0.

Traverse the linked list using `curr` while incrementing `count` in each iteration.

When `count` becomes equal to `k`, we know that we've reached the kth node from the start. Save this node in `front`. In addition, set the `end` pointer at the head of the linked list.

Continue traversing the linked list by moving both `end` and `curr` forward. When `curr` reaches the last node, `end` reaches the kth node from the end of the linked.

Swap the values of the `front` and `end` nodes.

Return the head of the linked list.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in `SwapNodes.java` in the following coding playground. You'll need the provided supporting code to implement your solution. Some useful code templates have been provided that you may build on and use to solve this problem.

SwapNodes.java

LinkedListNode.java

LinkedList.java

LinkedListTraversal.java

LinkedListReversal.java

```
import java.util.*;
public class SwapNodes{
    public static LinkedListNode swapNodes(LinkedListNode head, int k) {
    // Write your code here
    return head;
  }
}
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

[3,2,3,4,5,6]

Input #2

3

Mark as Completed