TIPS Interviewer Packet B - Session 3

# Interviewer:

## Behavioral:

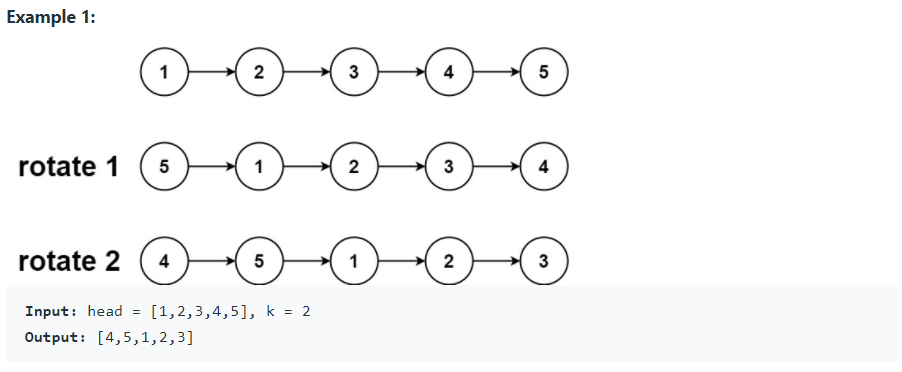
Tell me about a time when you had to lead a project. What did you learn from it?

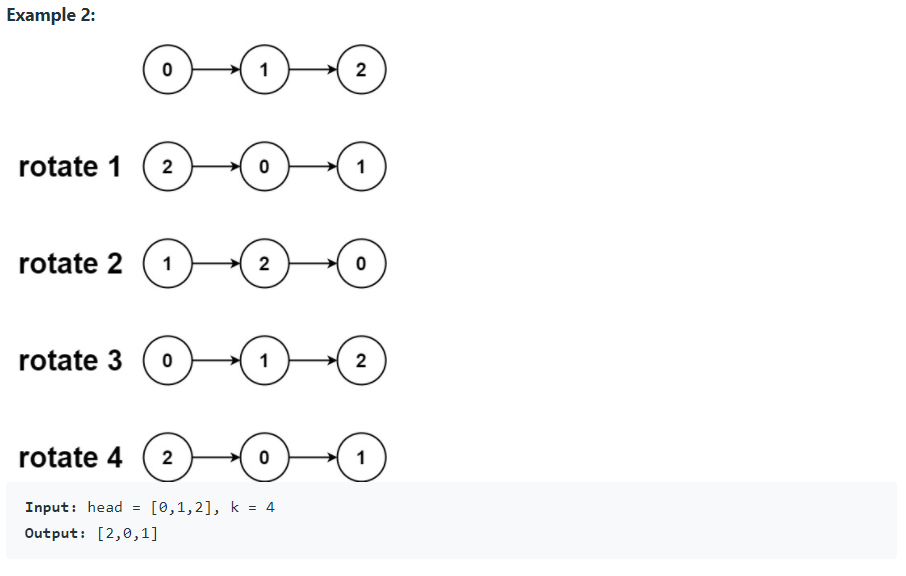
## Question:

<https://leetcode.com/problems/rotate-list/>

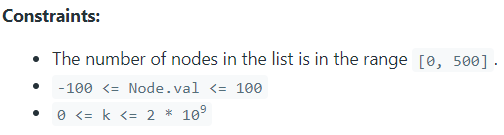
Given the head of a linked list, rotate the list to the right by k places.

## Examples:





## Follow up Q&A:



* Can the list be empty?
  + Yes. In that case, just return nullptr.
* Will k ever be negative?
  + No.
* What if k is greater than the size of the list?
  + See example 2. If k is greater than or equal to the size of the list n, then rotating by k nodes is the same as rotating by k mod n (k % n) nodes.

## Hint(s):

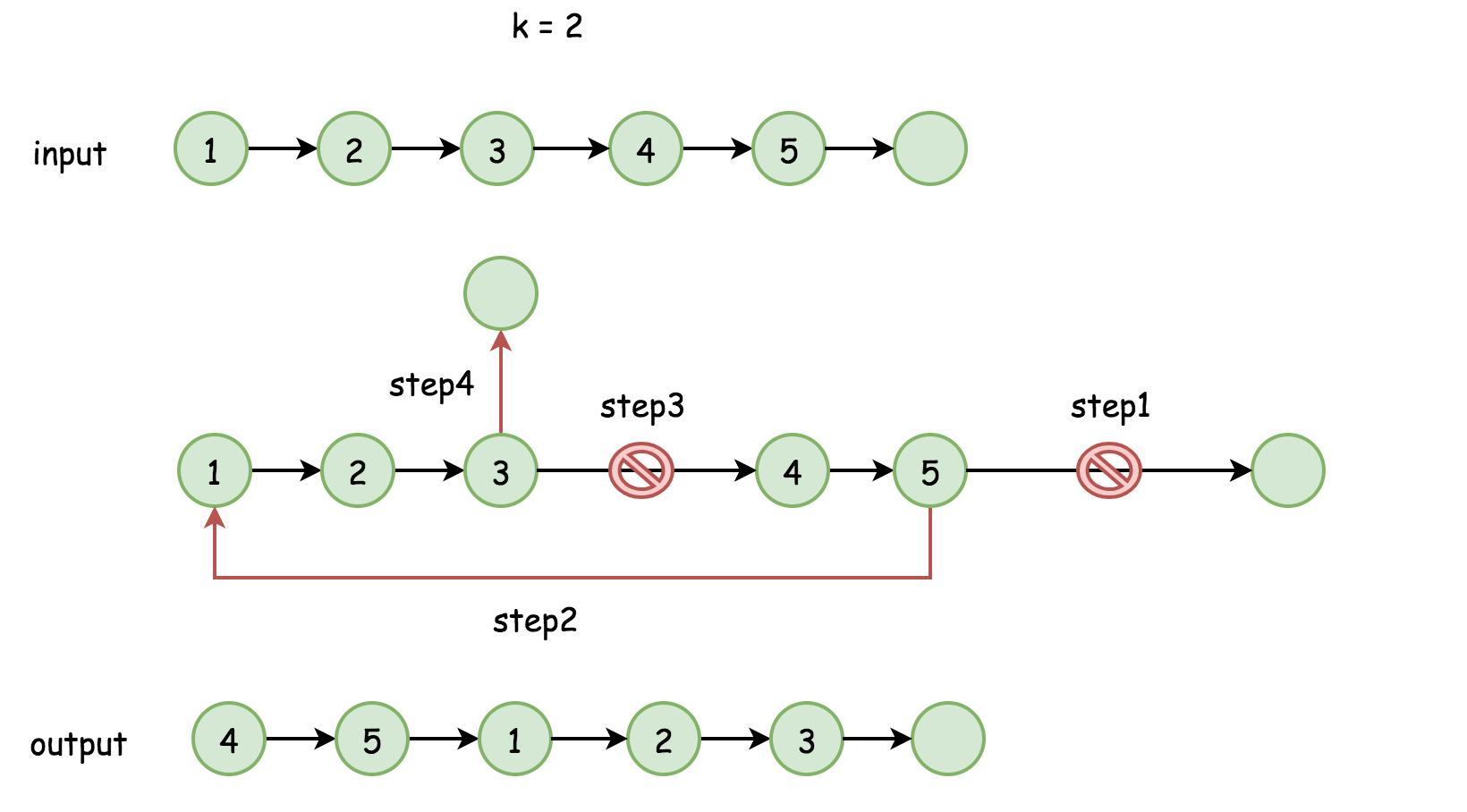
*Ask if they would like a hint before giving a hint*

* Could you perform the rotation more easily by making the last node in the list point back to the first node, making this a circularly linked list?
* If you look at the examples, it seems like rotating the lists is the same thing as splitting the list into two parts and moving the second part before the first part.

## Solution(s): (General concept and time/space complexity)

The nodes in the list are already linked, and hence the rotation basically means

* To close the linked list into a ring.
* To break the ring after the new tail and just in front of the new head.



Where is the new head?

In the position n - k, where n is the number of nodes in the list. The new tail is just before, in the position n - k - 1.

We were assuming that k < n. What about the case of k >= n?

k could be rewritten as a sum k = (k // n) \* n + k % n, where the first term doesn't result in any rotation. Hence one could simply replace k by k % n to always have a number of rotation places smaller than n.

Algorithm

The algorithm is quite straightforward :

* Find the old tail and connect it with the head old\_tail.next = head to close the ring. Compute the length of the list n at the same time.
* Find the new tail, which is (n - k % n - 1)th node from the head and the new head, which is (n - k % n)th node.
* Break the ring new\_tail.next = None and return new\_head.

Implementation

1 / 14

Complexity Analysis

Time complexity: O(N) where N is the number of nodes in the list.

Space complexity: O(1).

### Other questions follow up

*Ask if there is more than 5 minutes remaining when they finish their code and testing.*

* How would you need to change your code to accommodate negative values of k? What would rotating right by a negative value even mean?
  + Conceptually, rotating right by -k would be like rotating left by k.
  + Adding/subtracting n to/from k produces an identical result, so if k is negative, you could add n to it until it becomes positive. For example, given k=-10 and n=4, rotating right by -10 + 3(4) = -10 + 12 = 2 would be equivalent.
  + So adding a check for a negative k and replacing it with an equivalent positive k would let this code generalize.
    - k < 0 ? k=ceil(-1\*k/n)\*n : k=k; would work!

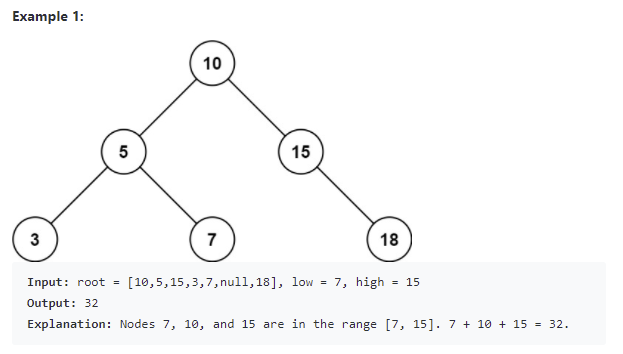
# Interviewee:

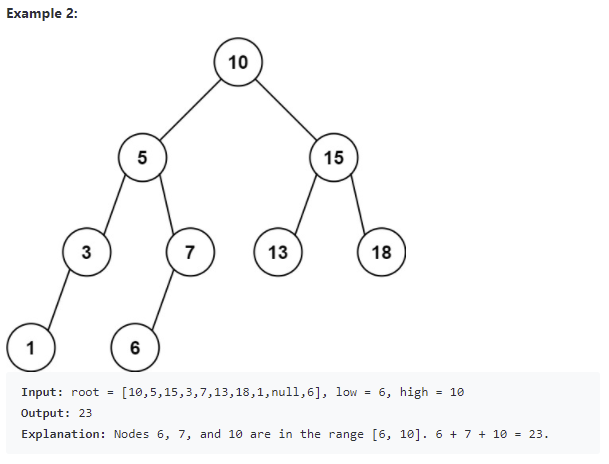
## Question:

<https://leetcode.com/problems/range-sum-of-bst/>

Given the root node of a binary search tree and two integers low and high, return *the sum of values of all nodes with a value in the inclusive range* [low, high].

## Examples:





## 

## Code below or on leetcode