Technical Question 5:

Explanation of code:

I believe an efficient method to solve this problem is to traverse the list once and store each value in a new array. We can use the array's built-in index property to accurately match the position of each node to the array element.

We begin with the first node and cycle through each succeeding one. We know when we reach the end because the node’s .next property will be set to None. We set that as our base condition.

We finally reach the end and exit the loop, storing the length. We then iterate over the nodes again, and this time stopping at our target node which is length – m + 1. We return that node.

Time efficiency of this is O(2n) since we traverse the list twice. Space efficiency is negligible since we are only using small variables.