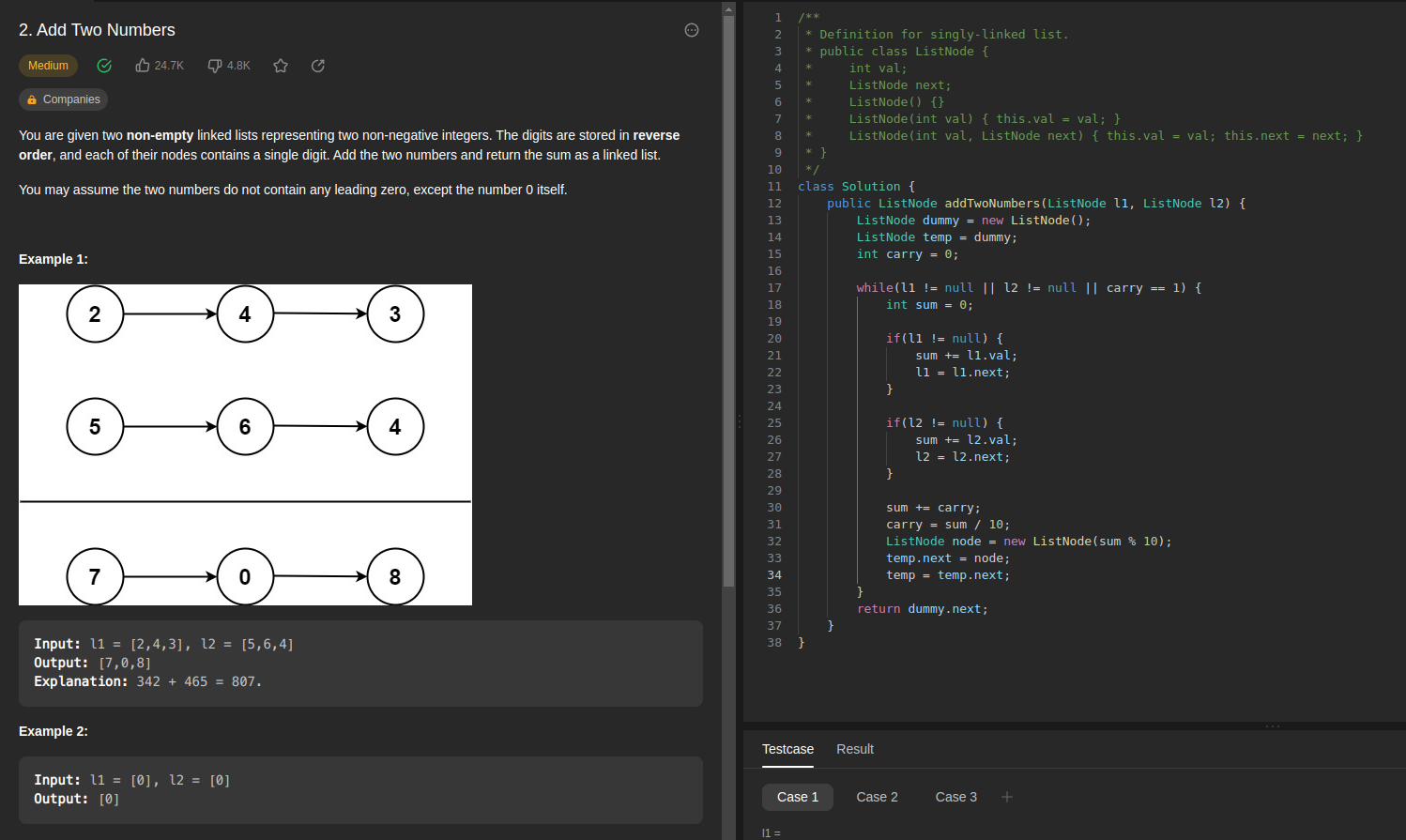
|  |  |
| --- | --- |
| **LINKED LISTS** | |
| Add two numbers | 1 |
| Palindrome Linked list | 1 |
| Linked List cycle | 2 |
| Linked List cycle II | 2 |
| Remove Nth node from the last | 3 |
| Merge K sorted lists | 3 |
| Design browser history (Doubly linked list) | 4 |
|  |  |
|  |  |

**Add two numbers:**

TC: O(N) // N is length of longest linked list

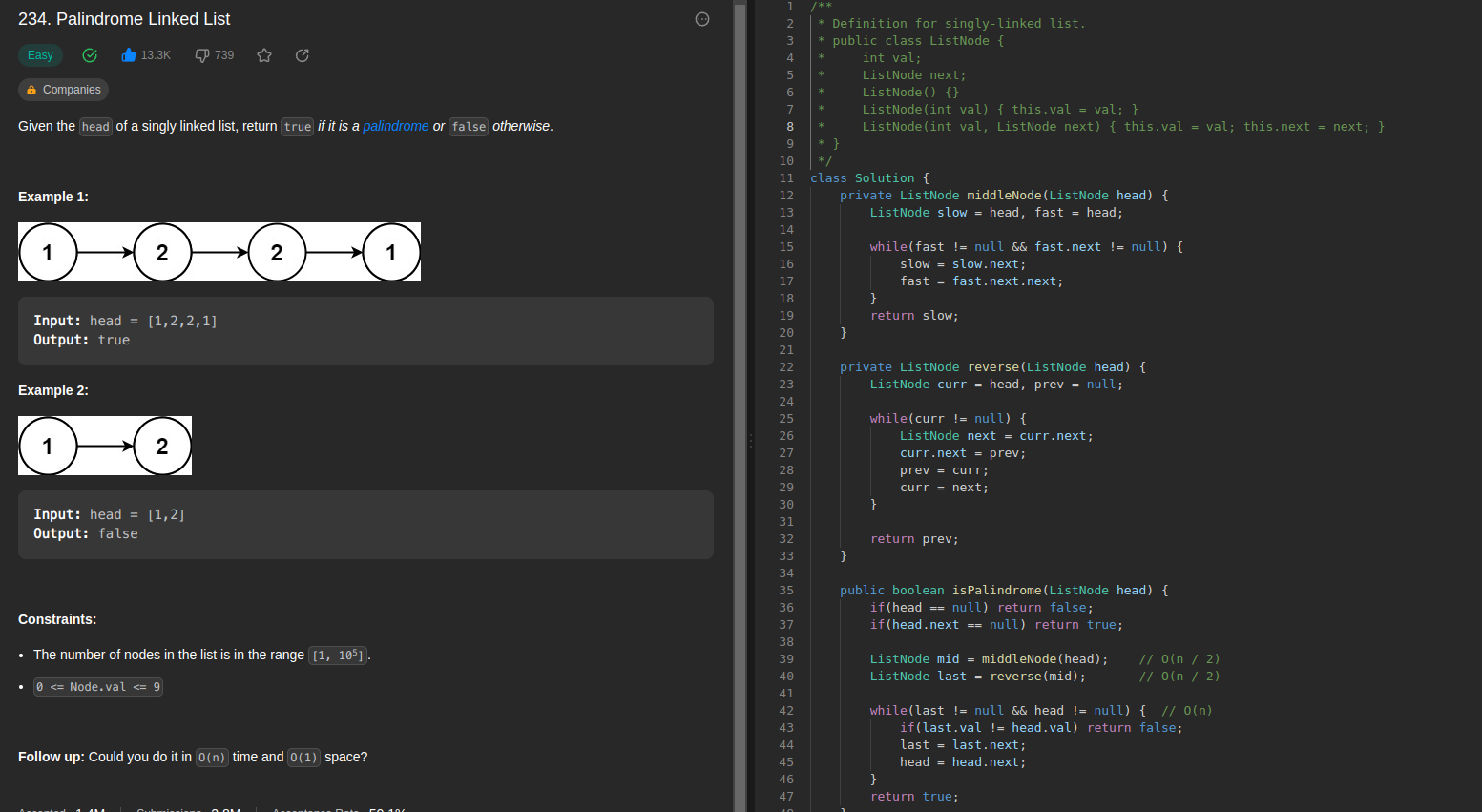
SC: O(1) // If we don’t consider the resultant linked list



**Palindrome linked list**

TC: O(N)

SC: O(1)

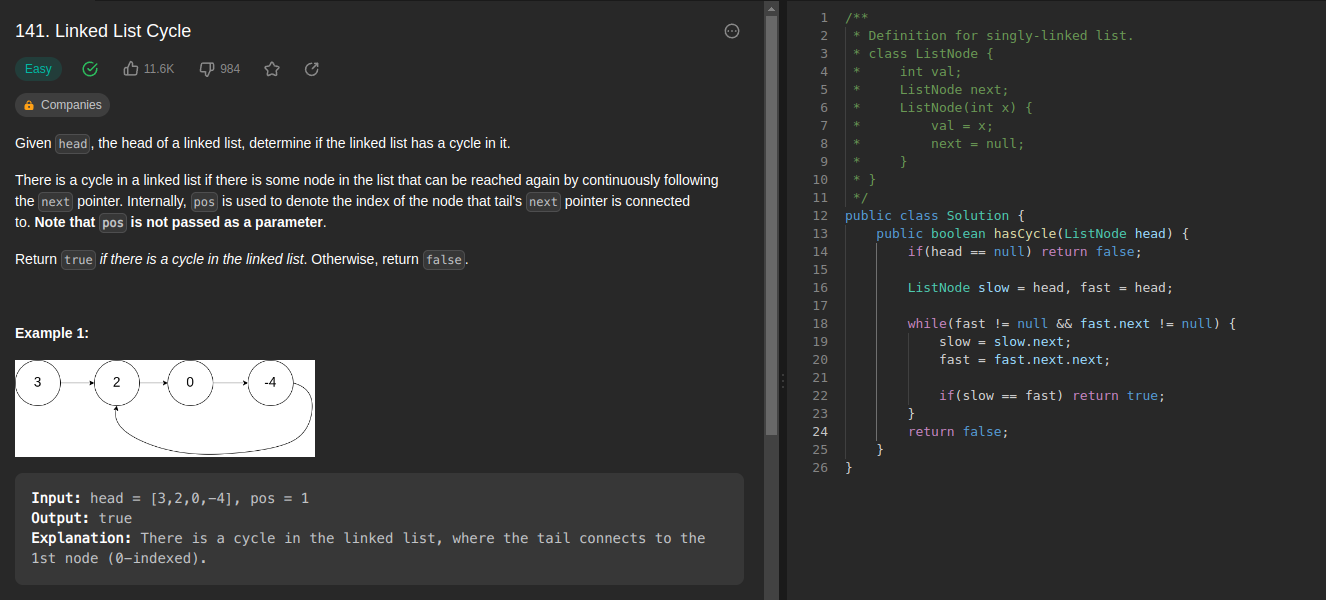
****

**Linked list Cycle:**

TC: O(N)

SC: O(0)

BF: push all Nodes in to hashset/hashmap. If the node already exists then there is a cycle

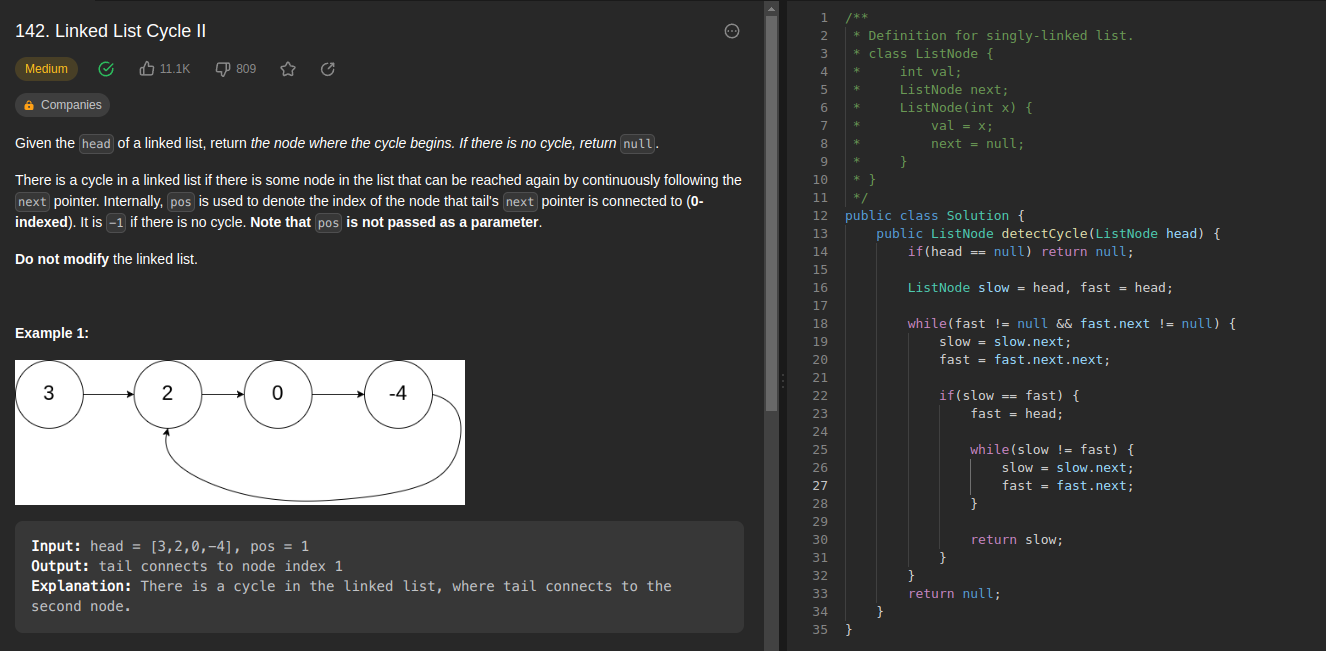


**Linked list Cycle II:**

TC: O(N)

SC: O(0)

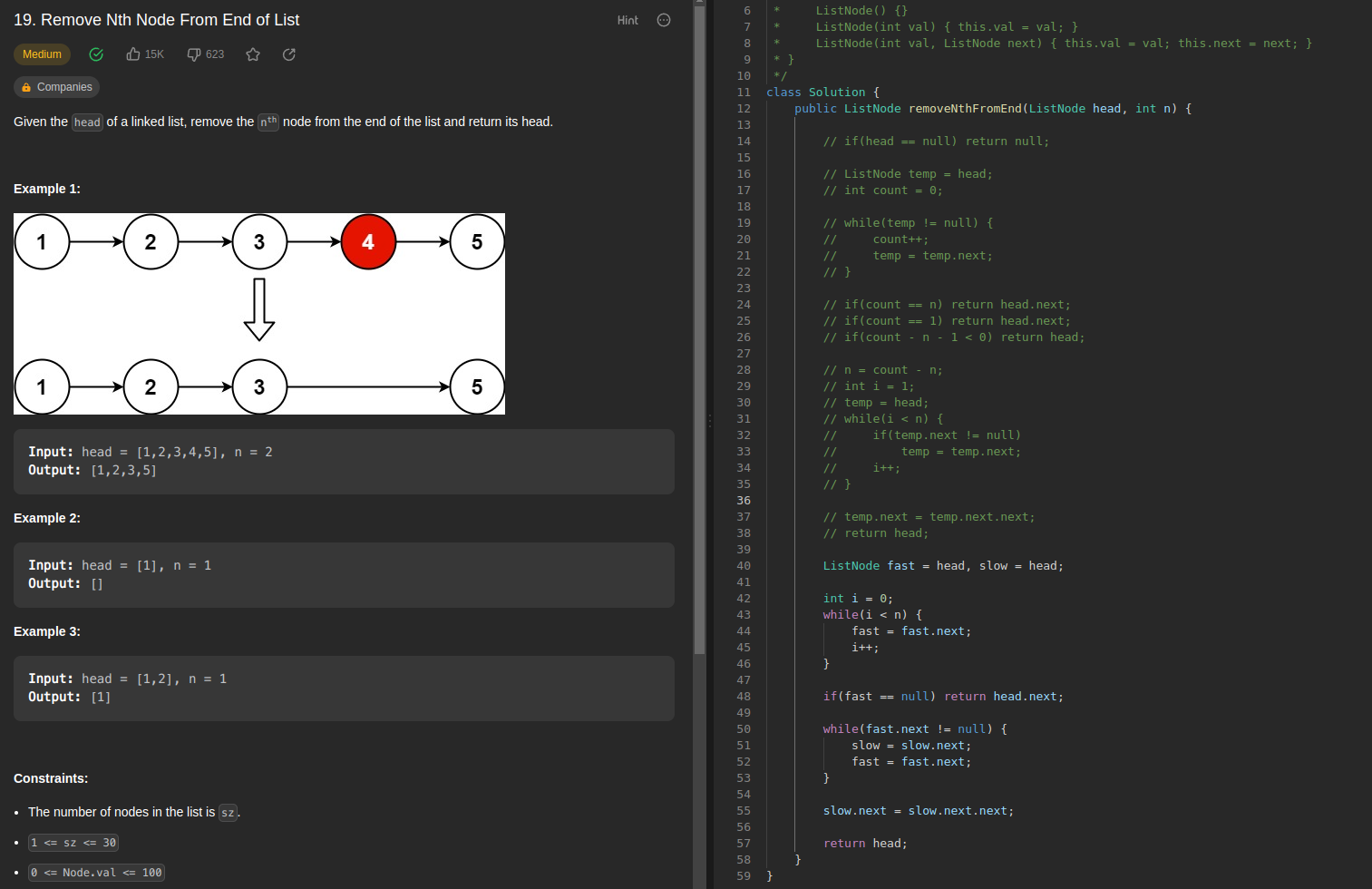
BF: Same as the above problem but return the node if it already exists in a hashmap / hashset.



**Remove Nth node from the last:**

TC: O(N)

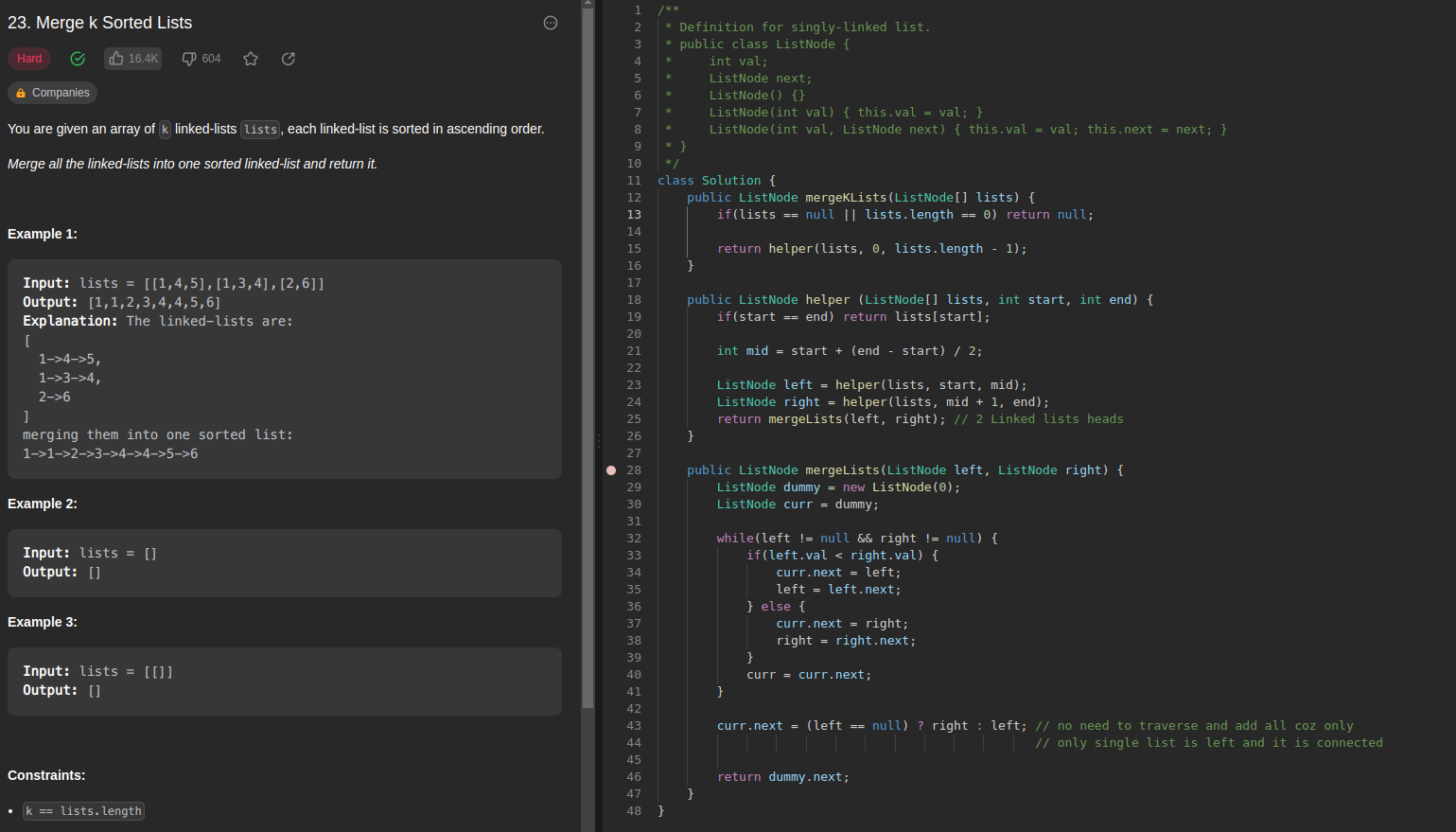
SC: O(1)



**Merge K sorted lists**

TC: O(N)

SC: O(1)



**Design browser history (Doubly linked list):**

TC: O(N)

SC: O(N)

