Linked List Cycle

Try to solve the Linked List Cycle problem.

We'll cover the following

Statement

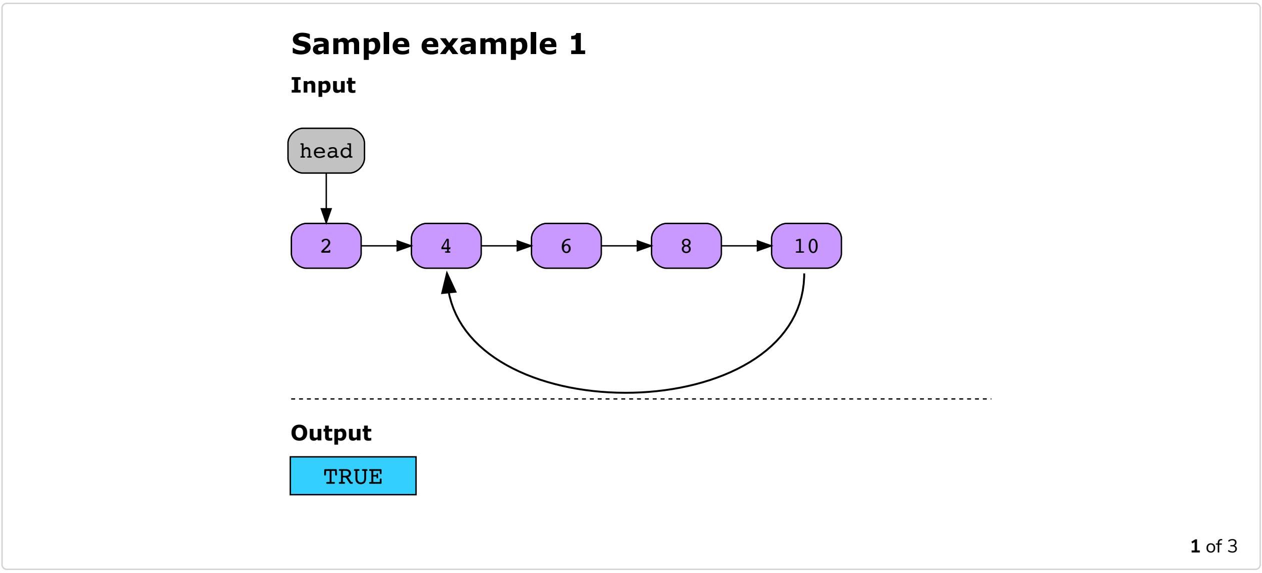
Check whether or not a linked list contains a cycle. If a cycle exists, return TRUE. Otherwise, return FALSE. The cycle means that at least one node can be reached again by traversing the `next` pointer.

Constraints:

Let `n` be the number of nodes in a linked list.

- $0 \leq$ `n` ≤ 500
- $-10^5 \leq$ `Node.data` $\leq 10^5$

Examples

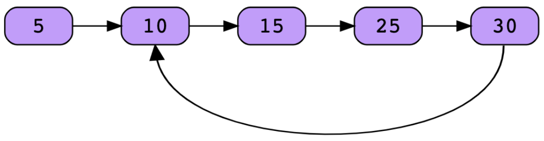


Understand the problem

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:

Linked List Cycle

1 What is the output if the following linked list is given as input?



A) TRUE

B) FALSE

Submit Answer

Question 1 of 3
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 both fast and slow pointers to the head of the linked list.

Move the slow pointer one node ahead and the fast pointer two nodes ahead.

Check if both pointers point to the same node at any point. If yes, then return TRUE.

Else, if the fast pointer reaches the end of the linked list, return FALSE.

Reset Show Solution Submit

Try it yourself

Implement your solution in `CycleDetection.java` in the following coding playground. You'll need the provided supporting code to implement your solution.

Java

LinkedList.java

LinkedListNode.java

```java
    public static boolean detectCycle(LinkedListNode head) {
    // Write your code here

    return false;
  }
}
```

Powered by AI Submit

Test Cases Results

Case 1 Case 2 Case 3

Input #1

[2,4,6,8,10]

Input #2

1

Linked List Cycle

> **Note:** The first input of the test case includes an array representing the contents of a linked list. The second input represents the index of the node to which the tail pointer is pointing. It will be -1, in case it is pointing to NULL. The second parameter is not passed in the function, it is just to add the cycle to the linked list.

Back
Solution: Happy Num...

Next
Solution: Linked List C...

Mark as Completed