

educative

← Back To Course Home

Start of LinkedList Cycle (medium) We'll cover the following

```
Problem Statement

    Try it yourself

Solution
  Code

    Time Complexity

    Space Complexity
```

Examples:

head

Problem Statement

cycle.

Given the head of a Singly LinkedList that contains a cycle, write a function to find the starting node of the

Cycle start

₩

? Ask a Question

```
Cycle start
                              head
                            head
                  Cycle start
Try it yourself
Try solving this question here:
            Python3
                                    ⊚ C++
                         JS JS
  👙 Java
      class ListNode {
        int value = 0;
```

ListNode(int value) { this.value = value;

ListNode next;

```
class LinkedListCycleStart {
   11
         public static ListNode findCycleStart(ListNode head) {
   12
          // TODO: Write your code here
   13
           return head;
   14
   15
   16
         public static void main(String[] args) {
  17
          ListNode head = new ListNode(1);
   18
   19
          head.next = new ListNode(2);
          head.next.next = new ListNode(3);
   20
           head.next.next.next = new ListNode(4);
   21
   22
           head.next.next.next.next = new ListNode(5);
   23
           head.next.next.next.next = new ListNode(6);
   24
  25
           head.next.next.next.next.next = head.next.next;
           System.out.println("LinkedList cycle start: " + LinkedListCycleStart.findCycleStart(head).value);
   26
  27
          head.next.next.next.next.next = head.next.next.next;
  28
   Run
                                                                                        Save
                                                                                                 Reset
Solution
If we know the length of the LinkedList cycle, we can find the start of the cycle through the following steps:
  1. Take two pointers. Let's call them pointer1 and pointer2.
  2. Initialize both pointers to point to the start of the LinkedList.
  3. We can find the length of the LinkedList cycle using the approach discussed in LinkedList Cycle. Let's
    assume that the length of the cycle is 'K' nodes.
  4. Move pointer2 ahead by 'K' nodes.
```

Let's visually see this with the above-mentioned Example-1:

6. As pointer2 is 'K' nodes ahead of pointer1, which means, pointer2 must have completed one loop in

the cycle when both pointers meet. Their meeting point will be the start of the cycle.

pointer1

-Keep incrementing both pointers until they meet

pointer1

5. Now, keep incrementing pointer1 and pointer2 until they both meet.

pointer1, pointer2

pointer1, pointer2

We can use the algorithm discussed in LinkedList Cycle to find the length of the cycle and then follow the

-Move pointer2 '4' nodes ahead

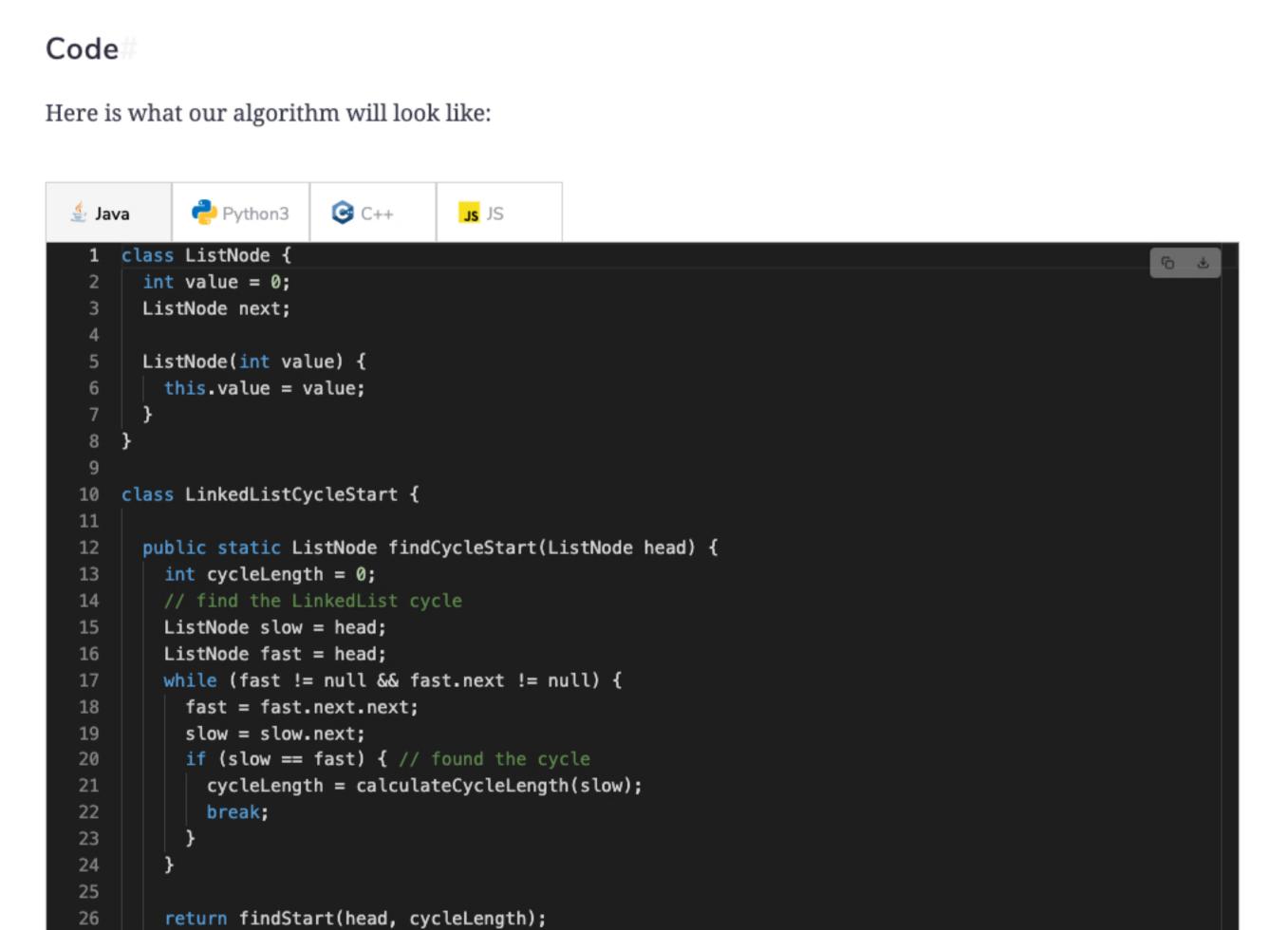
pointer2

pointer2

Save

() ¥

above-mentioned steps to find the start of the cycle.



The algorithm runs in constant space O(1).

Try it yourself

Solution

Code

Time Complexity

Space Complexity

Space Complexity

Time Complexity

overall time complexity of our algorithm will be O(N).

27

28

Run

for Coding Questions

Search Course

Problem Challenge 2

Problem Challenge 3

Pattern: Fast & Slow

LinkedList Cycle (easy)

Start of LinkedList Cycle

Solution Review: Problem

Solution Review: Problem

Challenge 1

Challenge 2

Challenge 3

Introduction

(medium)

pointers

18% completed

We'll cover the following Problem Statement

Cycle start

Cycle start

As we know, finding the cycle in a LinkedList with 'N' nodes and also finding the length of the cycle requires

O(N). Also, as we saw in the above algorithm, we will need O(N) to find the start of the cycle. Therefore, the

Problem Statement Given the head of a Singly LinkedList that contains a cycle, write a function to find the starting node of the cycle. Examples: head

head

head

Js JS

⊚ C++

Cycle start -

10 class LinkedListCycleStart { 11 12 public static ListNode findCycleStart(ListNode head) {

Try it yourself

👙 Java

8 }

Try solving this question here:

class ListNode {

int value = 0;

ListNode next;

Python3

ListNode(int value) {

this.value = value;

pointer1, pointer2

-Move pointer2 '4' nodes ahead

pointer1

-Keep incrementing both pointers until they meet

pointer1

```
13
          // TODO: Write your code here
          return head;
  14
  15
  16
  17
        public static void main(String[] args) {
          ListNode head = new ListNode(1);
  18
          head.next = new ListNode(2);
  19
          head.next.next = new ListNode(3);
  20
  21
          head.next.next.next = new ListNode(4);
  22
          head.next.next.next = new ListNode(5);
  23
          head.next.next.next.next = new ListNode(6);
  24
  25
          head.next.next.next.next.next = head.next.next;
  26
          System.out.println("LinkedList cycle start: " + LinkedListCycleStart.findCycleStart(head).value);
  27
  28
          head.next.next.next.next.next = head.next.next.next;
  Run
                                                                                       Save
                                                                                               Reset
Solution
If we know the length of the LinkedList cycle, we can find the start of the cycle through the following steps:
  1. Take two pointers. Let's call them pointer1 and pointer2.
  2. Initialize both pointers to point to the start of the LinkedList.
  3. We can find the length of the LinkedList cycle using the approach discussed in LinkedList Cycle. Let's
    assume that the length of the cycle is 'K' nodes.
  4. Move pointer2 ahead by 'K' nodes.
  5. Now, keep incrementing pointer1 and pointer2 until they both meet.
  6. As pointer2 is 'K' nodes ahead of pointer1, which means, pointer2 must have completed one loop in
    the cycle when both pointers meet. Their meeting point will be the start of the cycle.
Let's visually see this with the above-mentioned Example-1:
```

```
pointer1, pointer2
We can use the algorithm discussed in LinkedList Cycle to find the length of the cycle and then follow the
above-mentioned steps to find the start of the cycle.
Code
Here is what our algorithm will look like:
             Python3
                          ⊘ C++
                                      Js JS
  👙 Java
     class ListNode {
                                                                                                       6 4
        int value = 0;
        ListNode next;
         ListNode(int value) {
           this.value = value;
   10 class LinkedListCycleStart {
   11
```

pointer2

pointer2

public static ListNode findCycleStart(ListNode head) { 12 int cycleLength = 0; 13 14 // find the LinkedList cycle ListNode slow = head; 15 16 ListNode fast = head; while (fast != null && fast.next != null) { 17 18 fast = fast.next.next; 19 slow = slow.next; 20 if (slow == fast) { // found the cycle cycleLength = calculateCycleLength(slow); 21 22 break; 23 24 25 return findStart(head, cycleLength); 26 27 28 Run Save Reset Time Complexity As we know, finding the cycle in a LinkedList with 'N' nodes and also finding the length of the cycle requires O(N). Also, as we saw in the above algorithm, we will need O(N) to find the start of the cycle. Therefore, the overall time complexity of our algorithm will be O(N).

← Back LinkedList Cycle (easy)

Want to work at Google, Facebook, or Amazon? Get hired faster with anonymous mock

interviews conducted by senior engineers from those companies. Detailed feedback helps you

Space Complexity

prep. See how ①

The algorithm runs in constant space O(1).

```
Report an Issue
```

Happy Number (medium)

✓ Mark as Completed

 \times

Next →