



Start of LinkedList Cycle (medium)

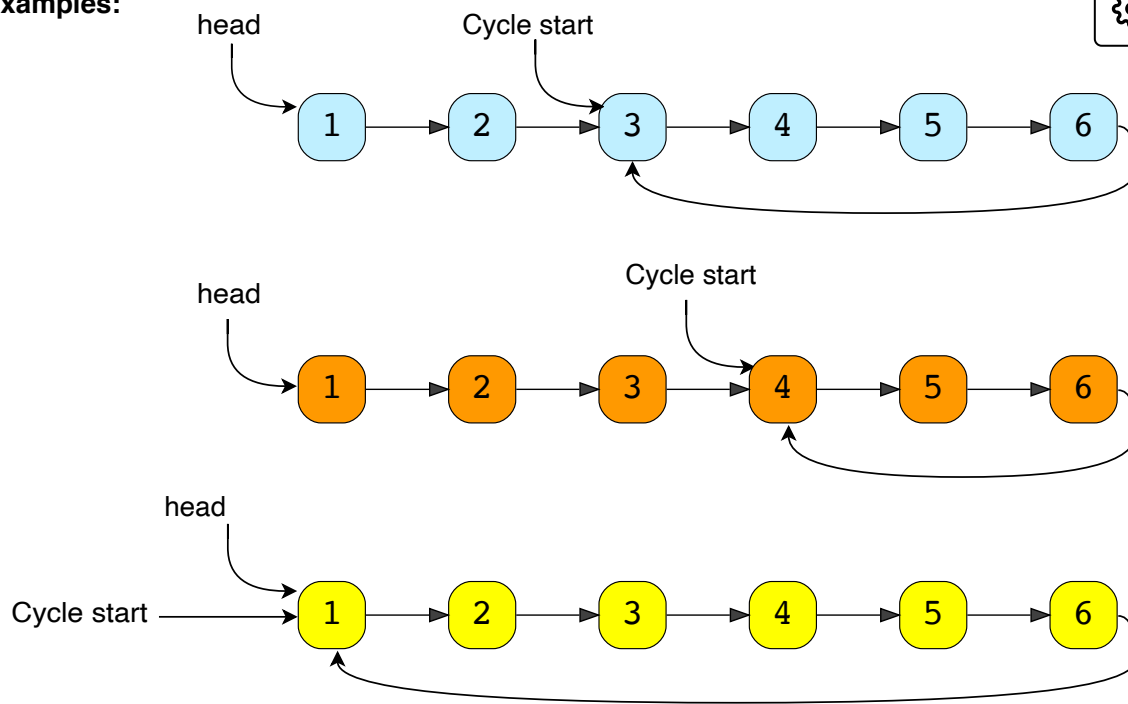
We'll cover the following



- Problem Statement
- Try it yourself
- Solution
 - Code
 - Time Complexity
 - Space Complexity

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:

Try it yourself

Try solving this question here:

Java

Python3

JS JS

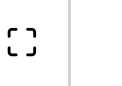
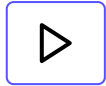
C++

```

22     fast = fast.next.next
23     slow = slow.next
24     if fast == slow:
25         length = cal_cycle_lenght(slow)
26         break
27     return find_start(head,length)
28
29 def cal_cycle_lenght(slow):
30     length = 1
31     curr = slow.next
32     while curr != slow:
33         curr = curr.next
34         length +=1
35     return length
36
37 def find_start(head,length):
38     p1,p2 = head,head
39     while length >0:

```

```
40     p2 = p2.next
41     length-=1
42     while p1 != p2:
43         p1 = p1.next
44         p2 = p2.next
45     return p1
46 def main():
47     head = Node(1)
48     head.next = Node(2)
49     head.next.next = Node(3)
```



Output

0.52s

```
LinkedList cycle start: 1
LinkedList cycle start: 1
LinkedList cycle start: 1
```

Solution

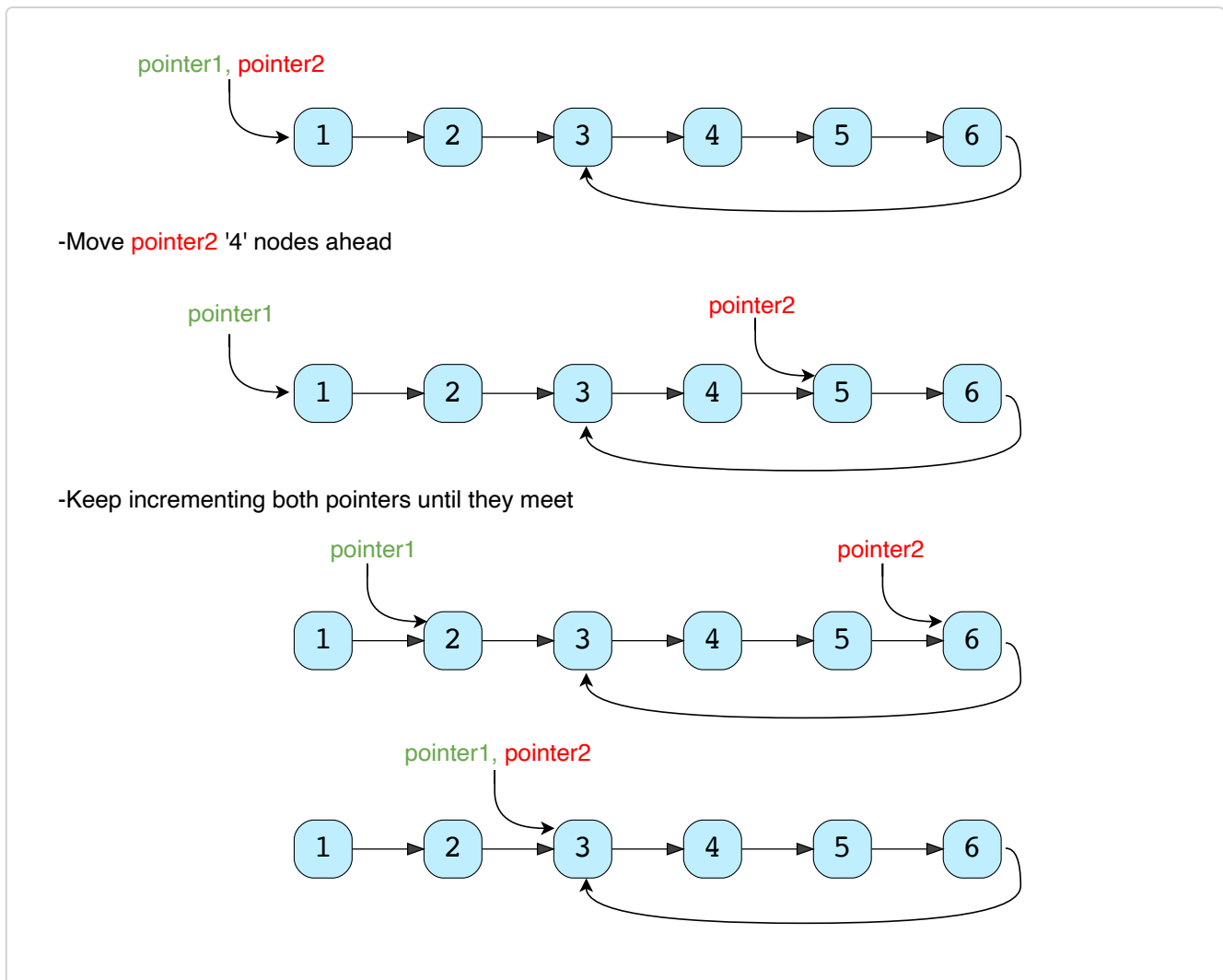
If we know the length of the **LinkedList** cycle, we can find the start of the cycle through the following steps:

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** (<https://www.educative.io/collection/page/5668639101419520/5671464854355968/6556337280385024>). 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:



We can use the algorithm discussed in [LinkedList Cycle](https://www.educative.io/collection/page/5668639101419520/5671464854355968/6556337280385024)

(<https://www.educative.io/collection/page/5668639101419520/5671464854355968/6556337280385024>) 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:

Java

Python3

C++

JS JS



```
20 slow, fast = head, head
21 while (fast is not None and fast.next is not None):
22     fast = fast.next.next
23     slow = slow.next
24     if slow == fast: # found the cycle
25         cycle_length = calculate_cycle_length(slow)
26         break
27 return find_start(head, cycle_length)
28
29
30 def calculate_cycle_length(slow):
31     current = slow
32     cycle_length = 0
33     while True:
34         current = current.next
35         cycle_length += 1
36         if current == slow:
37             break
38     return cycle_length
39
40 def find_start(head, cycle_length):
41     pointer1 = head
42     pointer2 = head
43     # move pointer2 ahead 'cycle_length' nodes
44     while cycle_length > 0:
45         pointer2 = pointer2.next
46         cycle_length -= 1
47     # increment both pointers until they meet at the start of the cycle
```



Output

0.17s

```
LinkedList cycle start: 3
LinkedList cycle start: 4
LinkedList cycle start: 1
```

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 $O(N)$.




Space Complexity

The algorithm runs in constant space $O(1)$.

[← Back](#)[Next →](#)[LinkedList Cycle \(easy\)](#)[Happy Number \(medium\)](#)☒ Mark as Completed

 Report
an Issue

 Ask a Question

(https://discuss.educative.io/tag/start-of-linkedlist-cycle-medium__pattern-fast-slow-pointers__grokking-the-coding-interview-patterns-for-coding-questions)