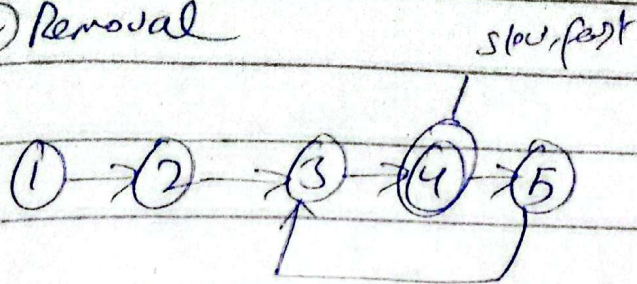


# Detect and Remove Loop Cycle in Linked List

① Detection

② Removal



① Detection

Loop  
(Never Ending)  
linked list

Slow-Fast Pointer approach

slow = head  $\rightarrow +1$

fast = head  $\rightarrow +2$

slow = head

fast = head

if loop exist

slowptr = fastptr

```
while (fast != NULL && fast->next != NULL) {
```

```
    slow = slow->next
```

```
    fast = fast->next->next
```

```
    if (slow == fast) return true;
```

```
    } return true;
```

② Detecting starting point of Loop

slow = head

```
while (slow != head)
```

```
    slow = slow->next; +1
```

```
    fast = fast->next; +1
```

```
    } return slow;
```

Removal part



```
listNode * prev = NULL;
```

```
prev = fast;
```

```
prev->next = NULL
```