

??

- 
- 

??


```

    @param head
    @return
    set
    set
    set
    +1
    O(N)
    O(n)

```

??????????????

```

    fast
    slow
    slow
    (
    )
    1
    (
    n
    c
    n=s+p
    c
    n
    c
    CB
    n-p=s
    s
    A
    B(AB=CB=s)

```

```

/**
 *
 * @param head
 * @return
 */
public static <T> ListNode<T> findEntranceInLoopList(ListNode<T> head){
    ListNode<T> slowPointer, fastPointer;

    //
    boolean isLoop = false;
    slowPointer = fastPointer = head;
    while(fastPointer != null && fastPointer.next != null){
        slowPointer = slowPointer.next;
        fastPointer = fastPointer.next.next;

        //
        if(slowPointer == fastPointer){
            isLoop = true;
            break;
        }
    }

    //
    if(isLoop){
        slowPointer = head;
        while(fastPointer != null && fastPointer.next != null){

```

```

//[]
if(slowPointer == fastPointer){
    return slowPointer;
}

slowPointer = slowPointer.next;
fastPointer = fastPointer.next;
}
}
return null;
}

```



$O(len1 * len2)$ []



hash []

hash



hash []

hash []



$O(\max(len1 + len2))$ []

$O(len1)$



Hash



Hash []



Hash []



$O(m+n)$ $m \leq n$



Hash []



$O(len1 + len2)$

K

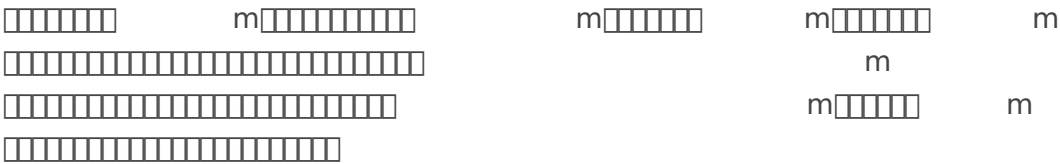
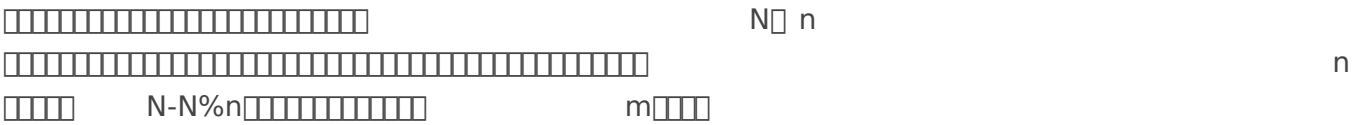
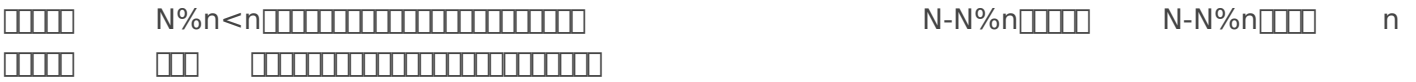


K []





????????



```
Node *getCircleHead(Node *node){
    //
    if (hasCircle(node)) {

        Node *fast=node;
        Node *slow=node;

        //
        m
        do{
            fast=fast->next->next;
            slow=slow->next;
        }while (fast!=slow);
```