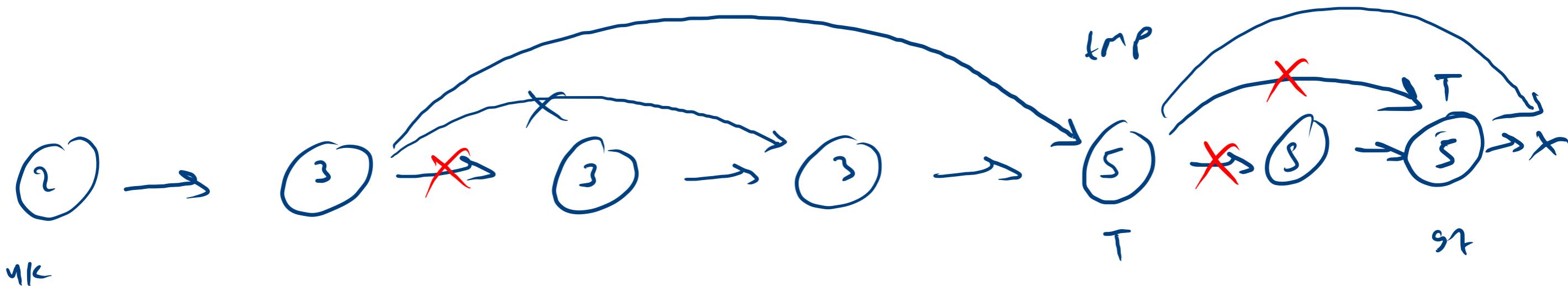
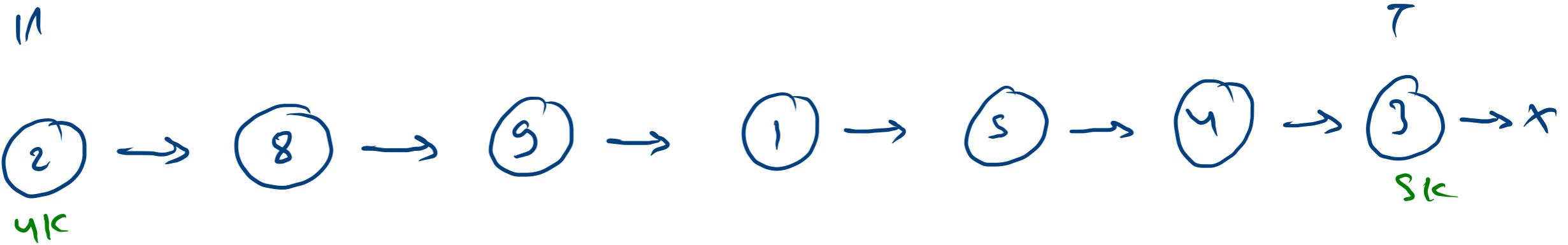


ans >

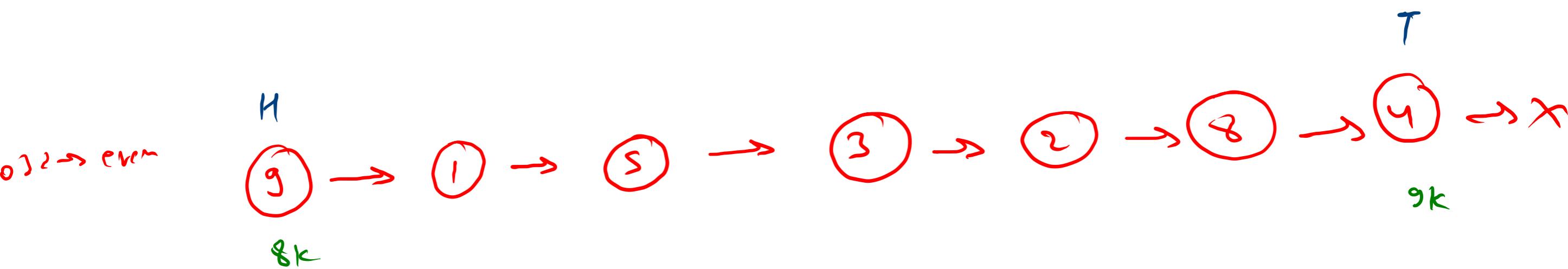
head	Q	tic
tail	X	stegic
size	Q	res

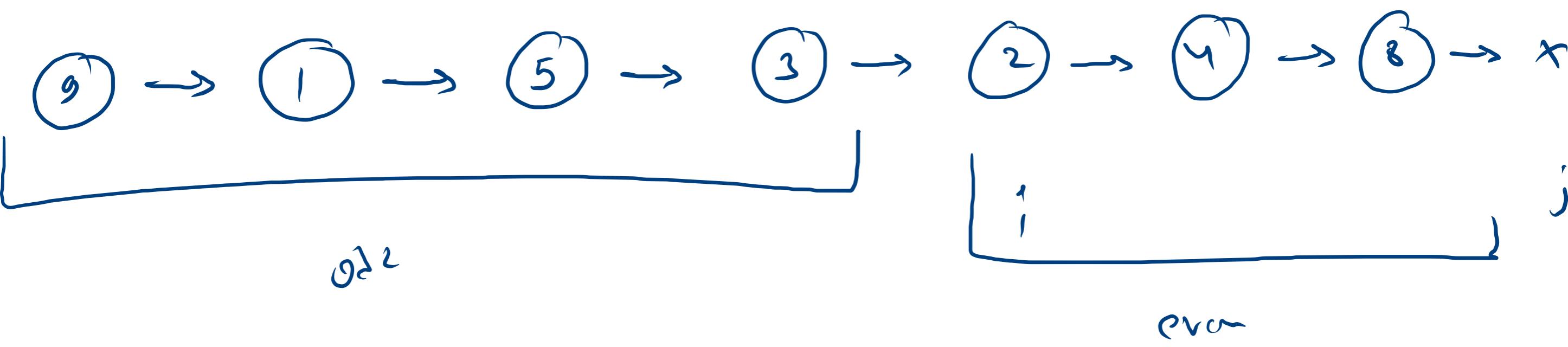


head	4lc
tail	9lc
size	3

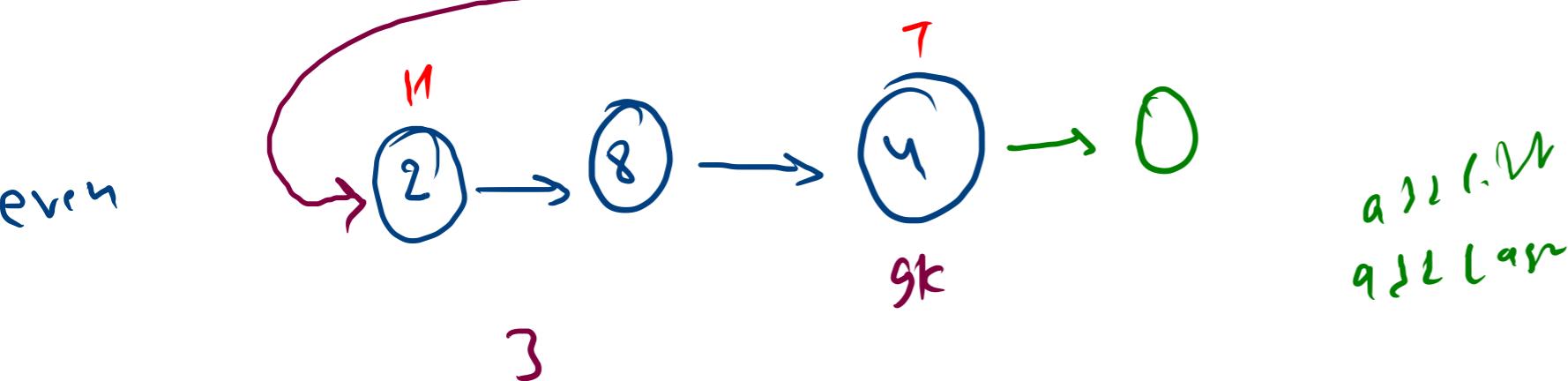
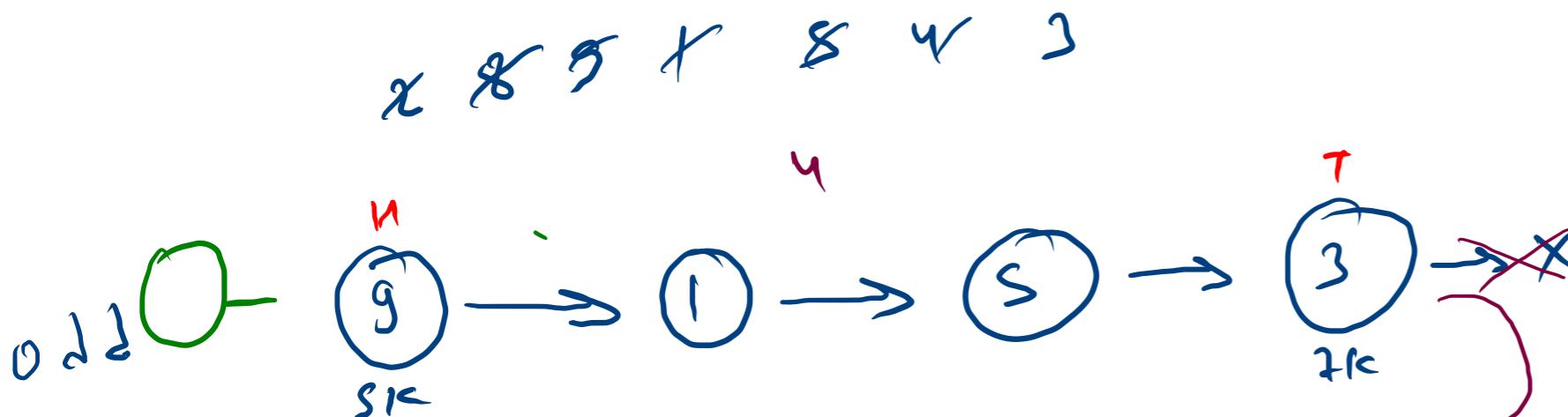
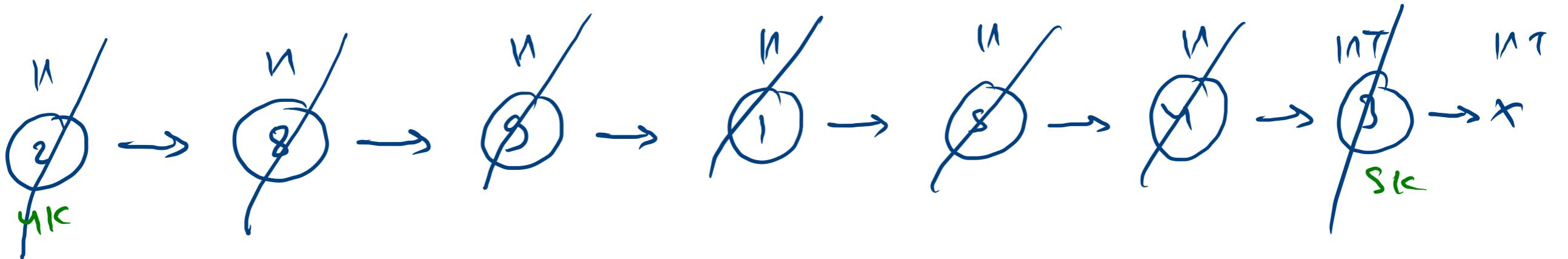


i j ↓





0/1 Segregation



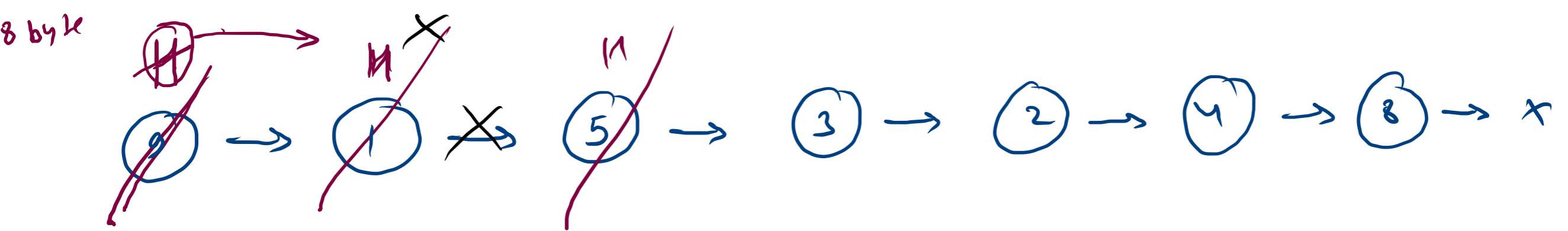
head
tail
size

head
tail
size

size
0?

$T(1 \rightarrow 3 5 2)$

$T(2 \rightarrow 2 8 4)$



`data = g[1]`

`while () {
 Non trp ;
}`

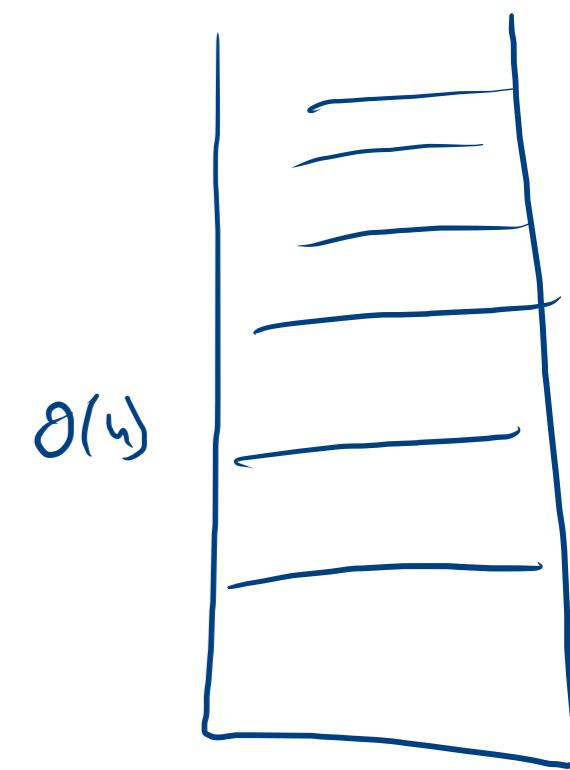
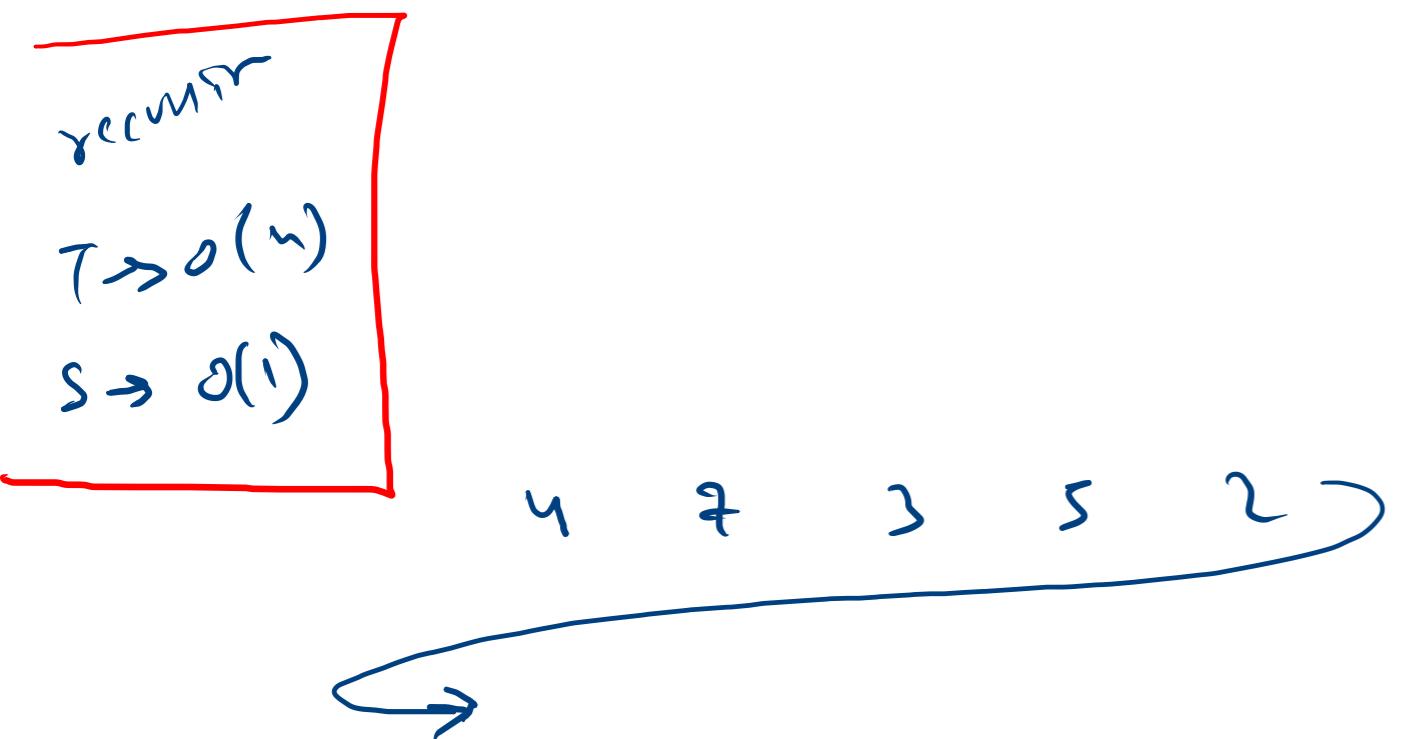
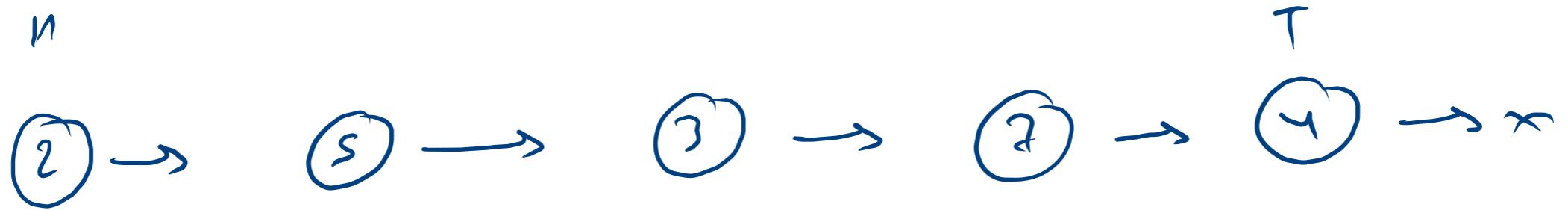
o} l

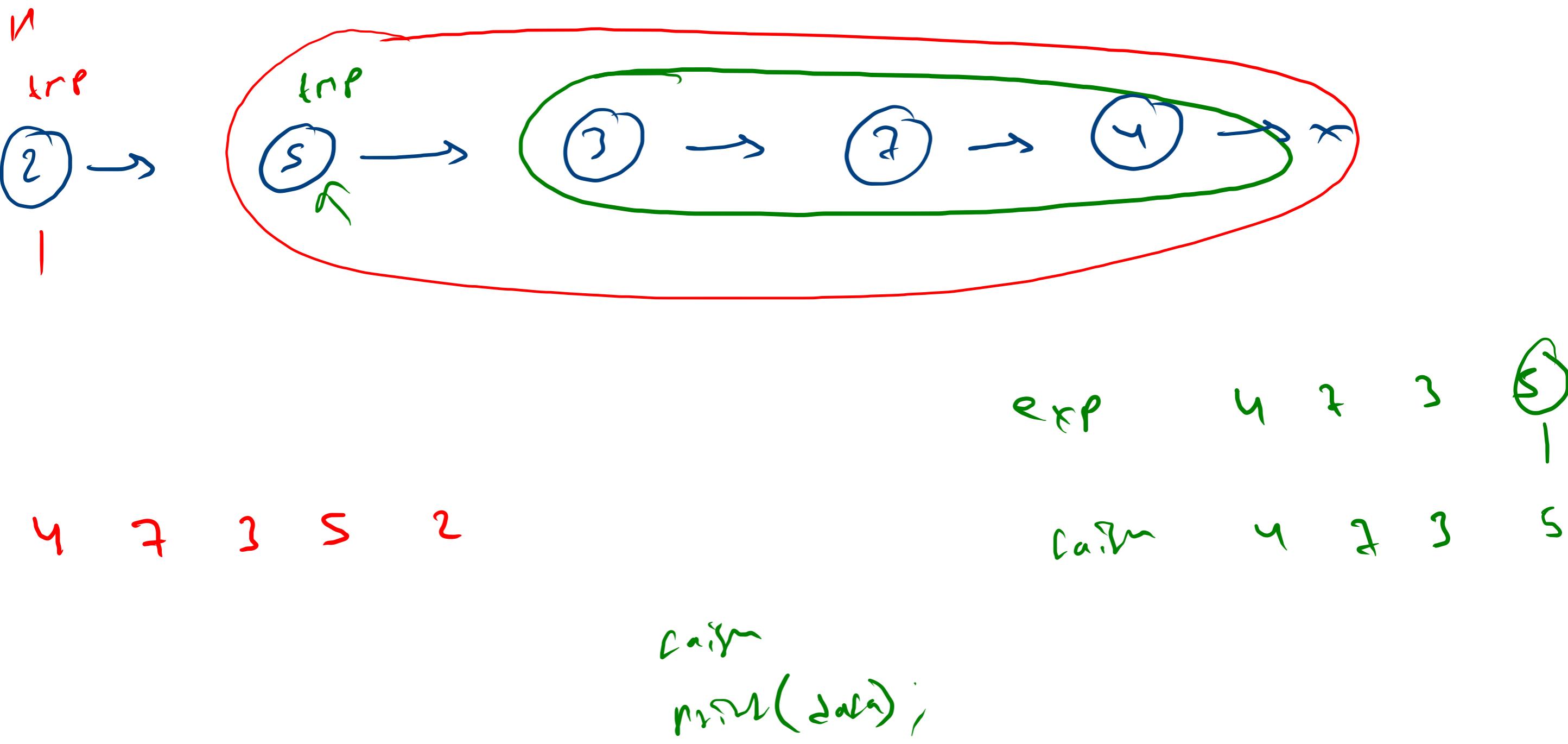


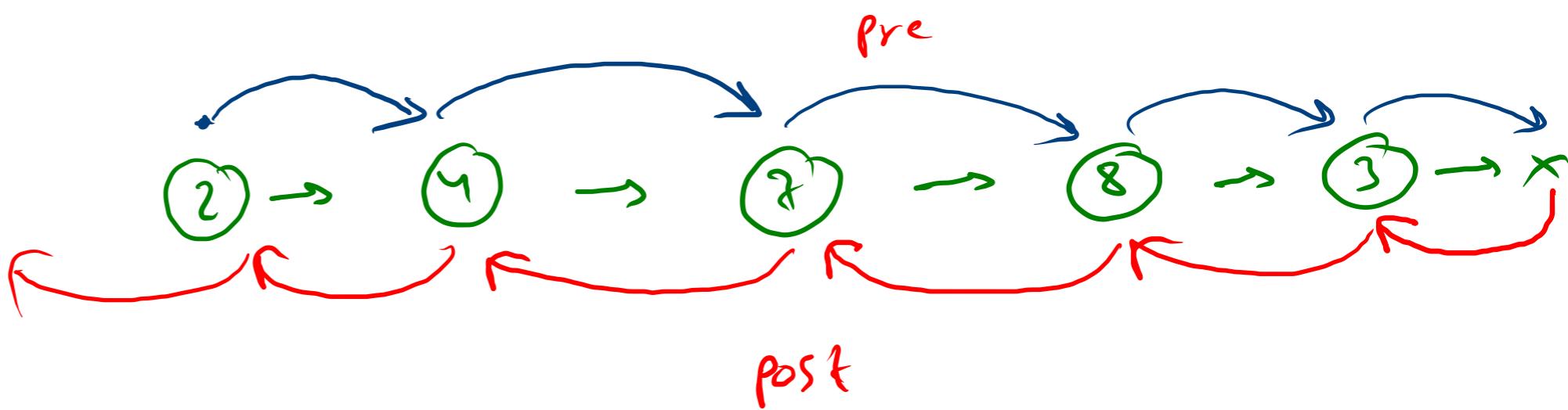
even

g
3

size
56 by 2
 48 by 2
 56 by 4
 48
 56

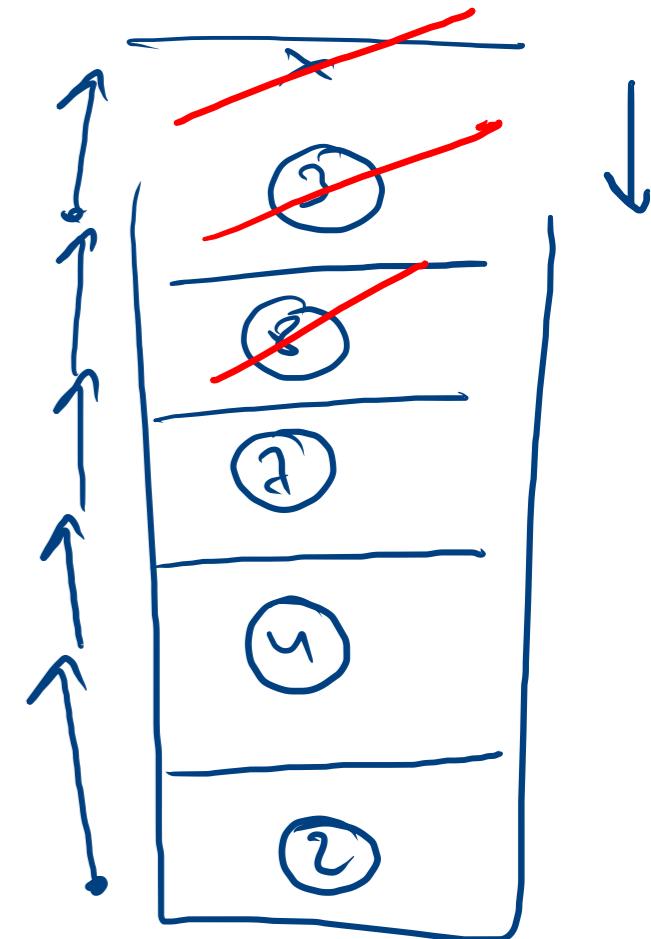


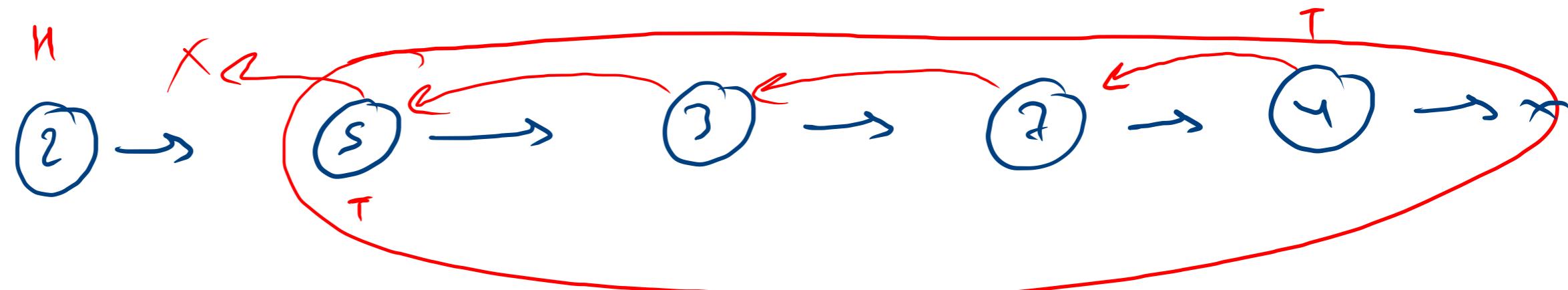




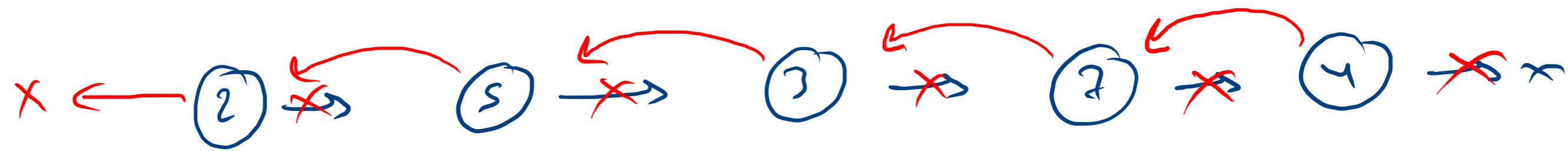
```

private void displayReverseHelper(Node node){
    if(node == null) return;
    pre
    displayReverseHelper(node.next);
    post
}
  
```



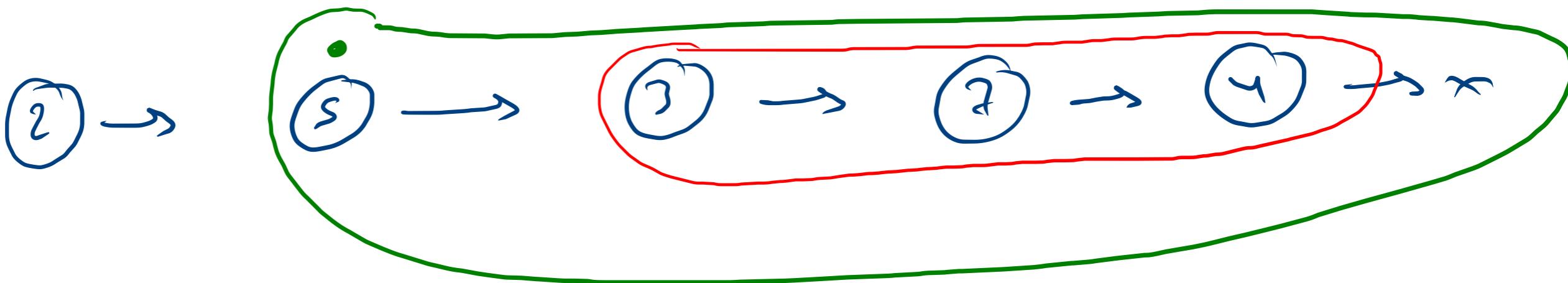


recursion

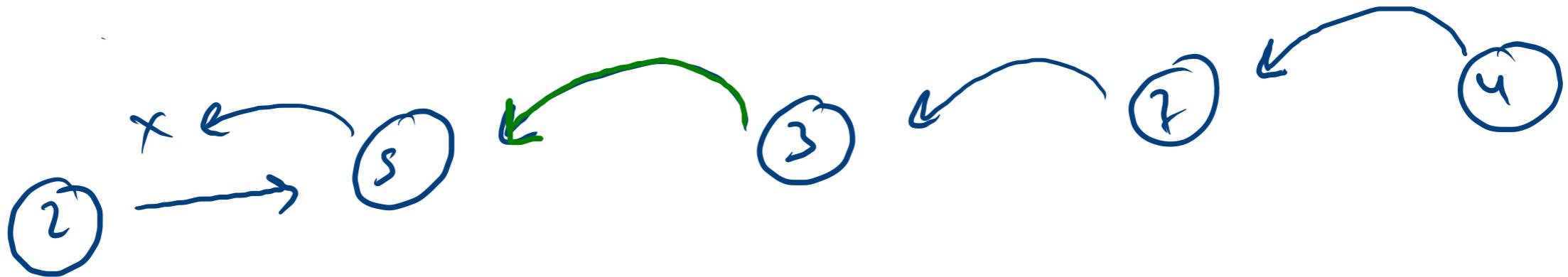


T

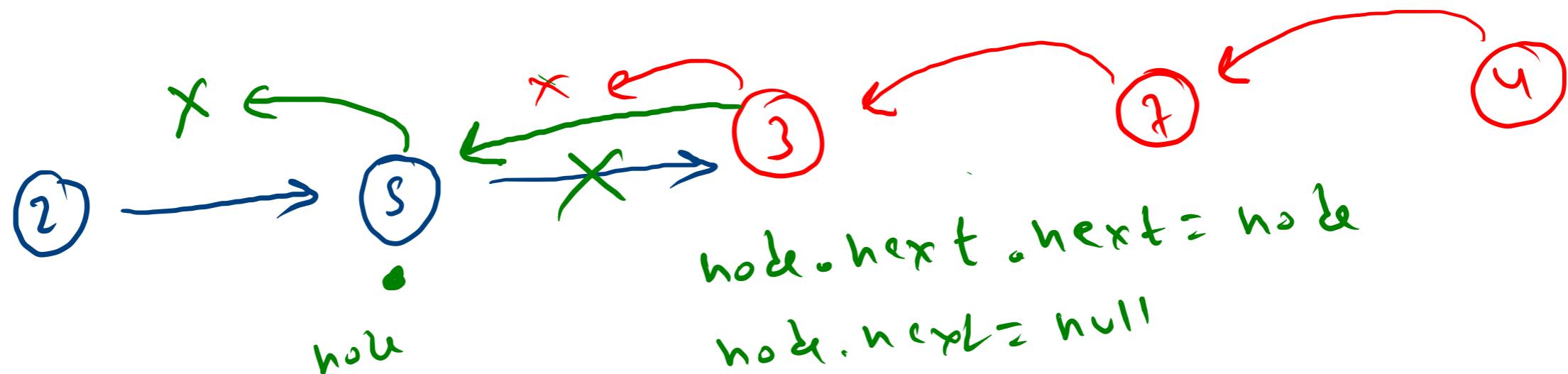
II

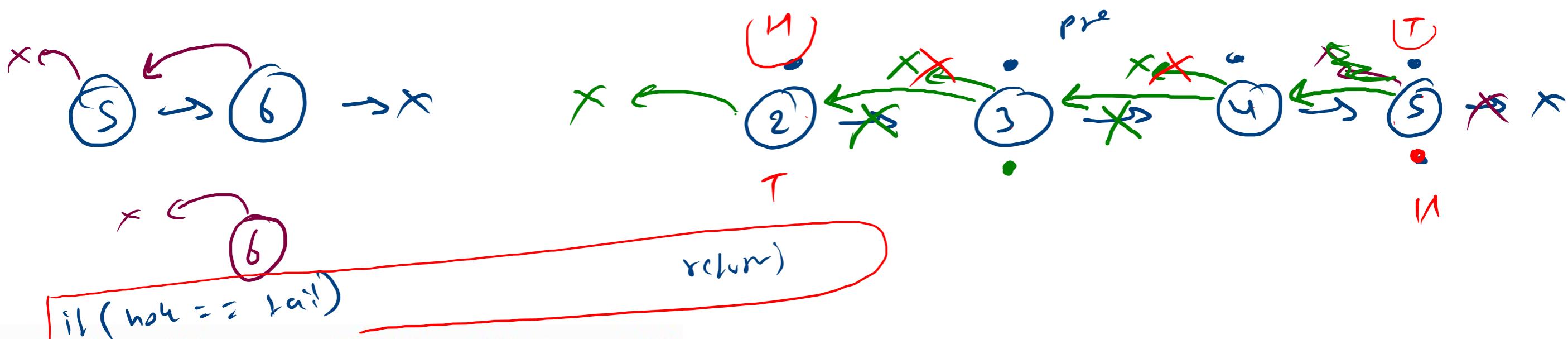


`expr` →



`chain`



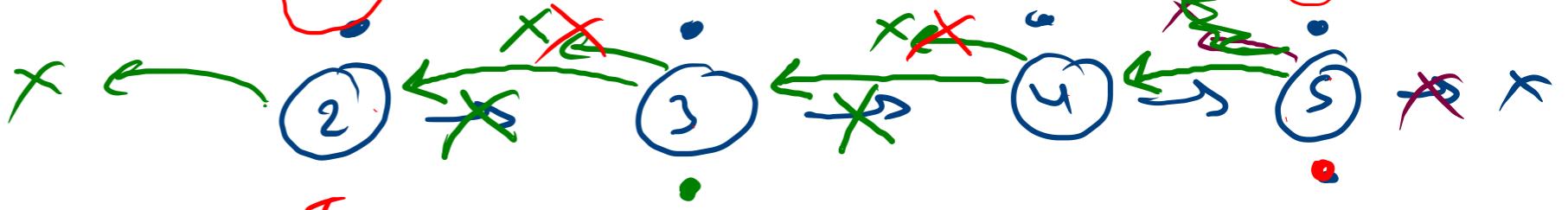


```

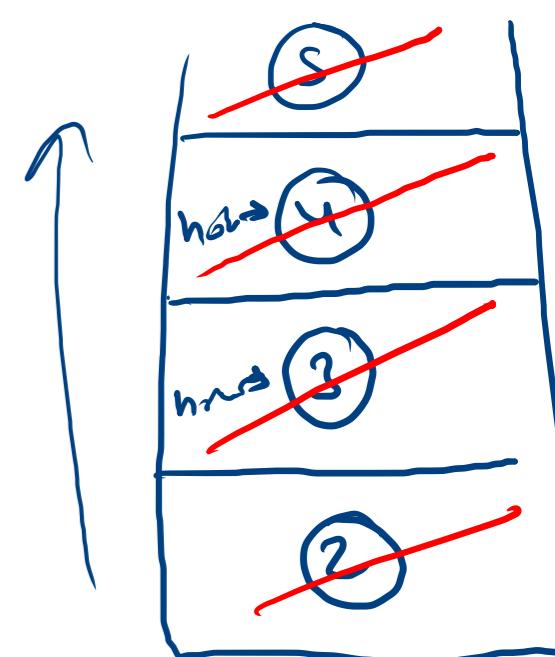
private void reversePRHelper(Node node){
    // write your code here
    reversePRHelper(node.next);
    node.next.next = node;
    node.next = null;
}

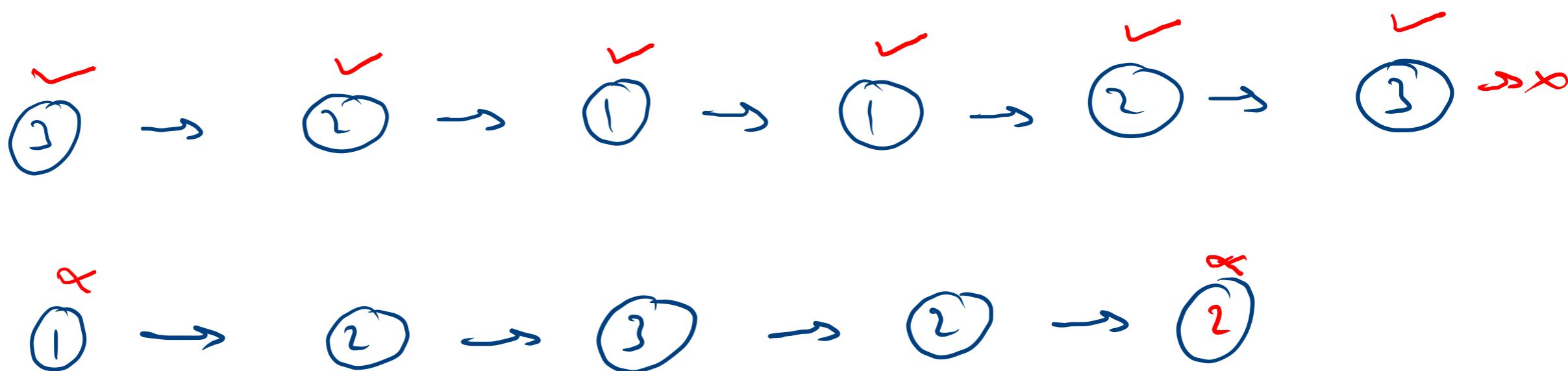
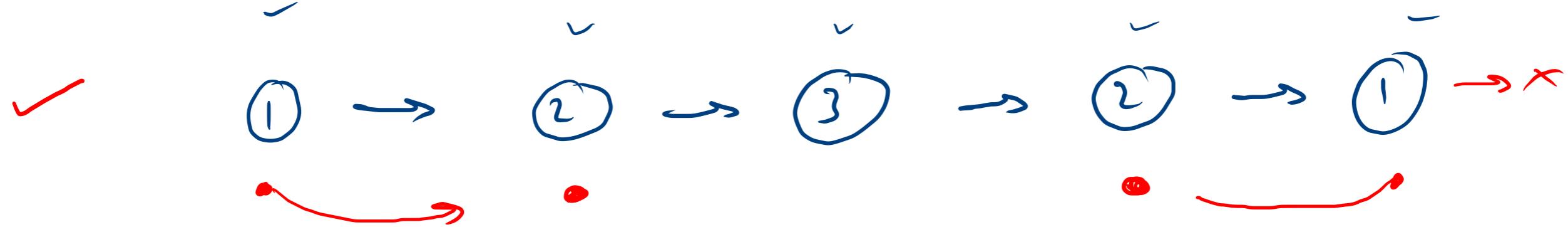
```

slice



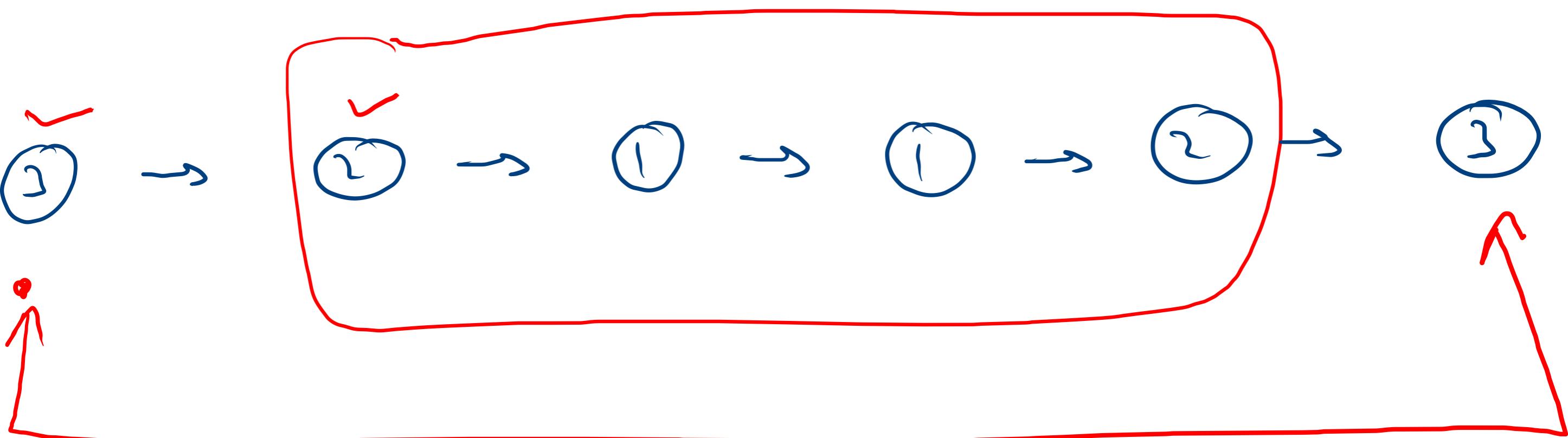
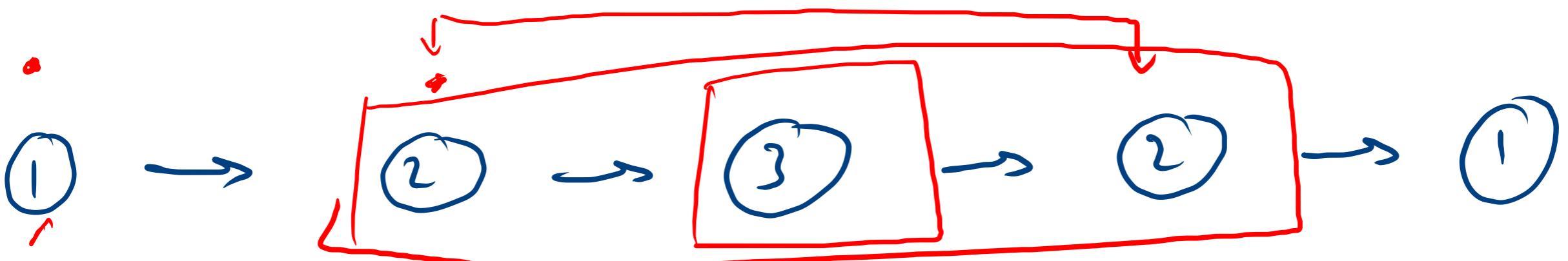
return

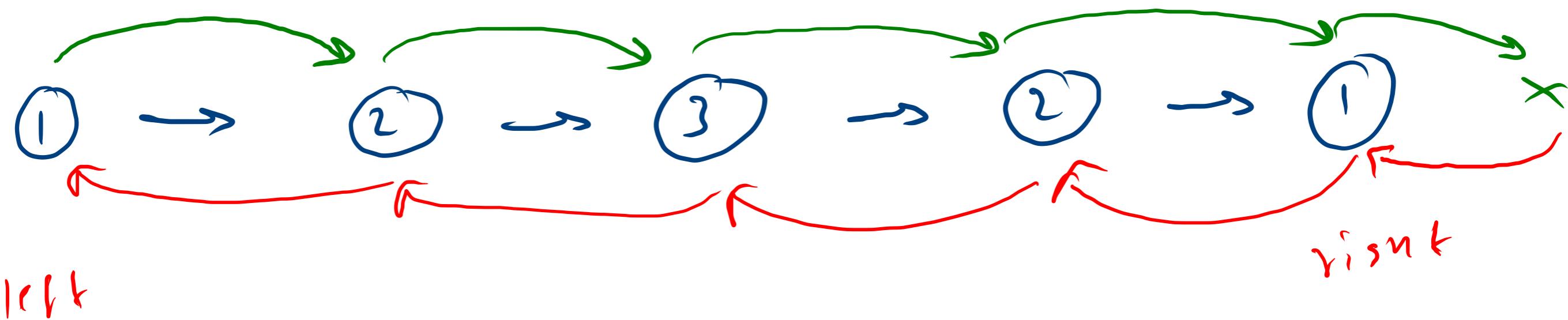


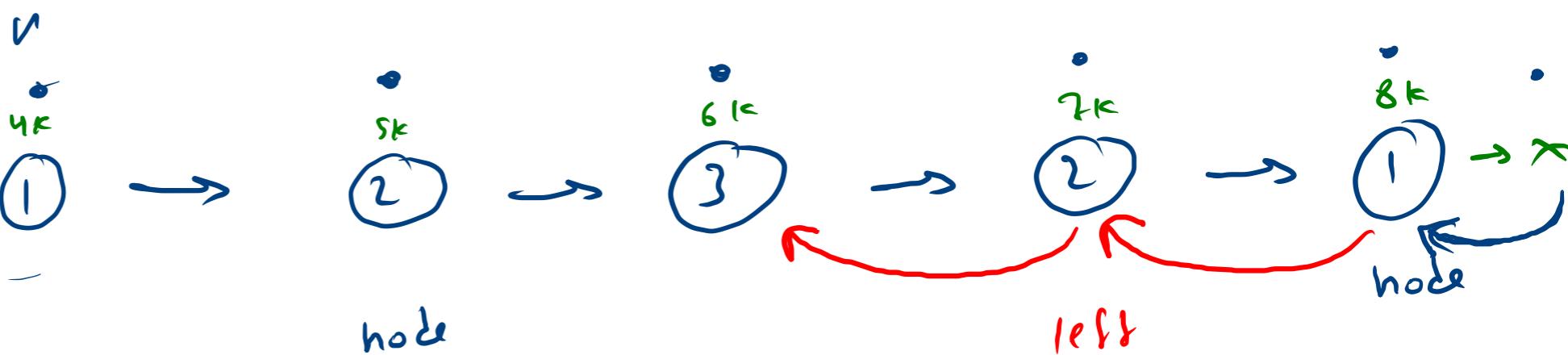


false

False





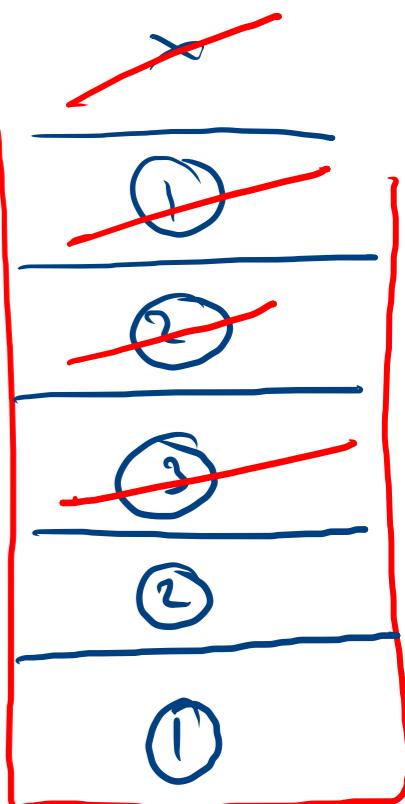


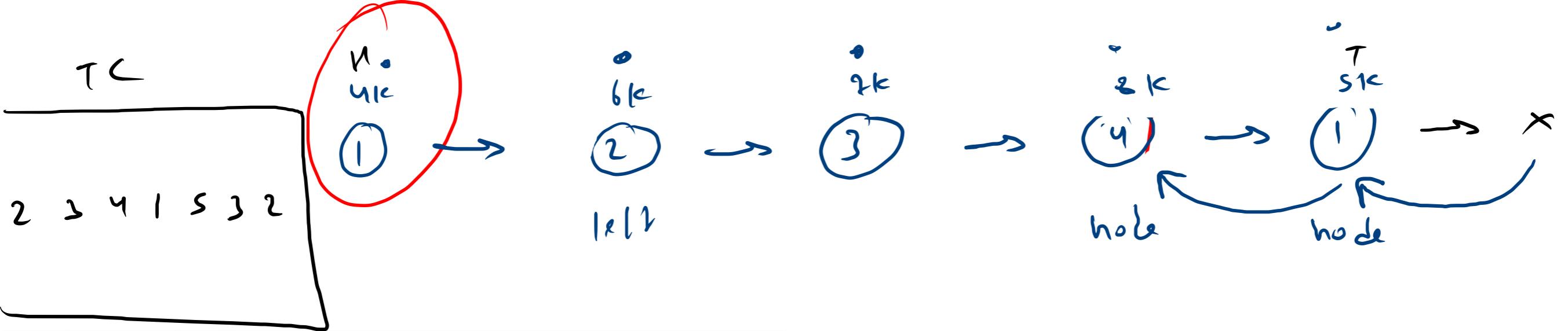
```

Node left=head;;
public boolean IsPalindromeHelper(Node node) {
    IsPalindromeHelper(node.next);
    left = left.next;
}

public boolean IsPalindrome() {
    IsPalindromeHelper(head);
}

```



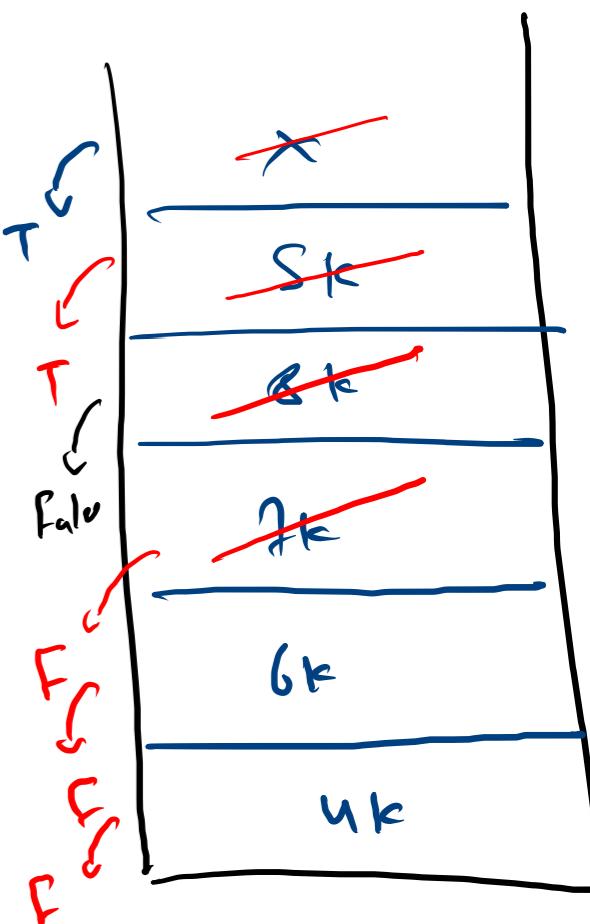
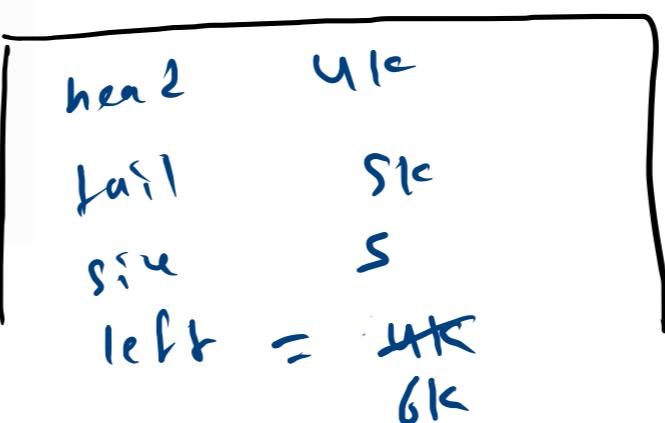


```

Node left=head;;
public boolean IsPalindromeHelper(Node node) {
    if(node == null) return true;

    boolean result = IsPalindromeHelper(node.next);
    if(result == false || node.data != left.data){
        return false;
    }
    left = left.next;
    return true;
}

```



H.V.

K Reverse In Linked List

YH:5

near

14

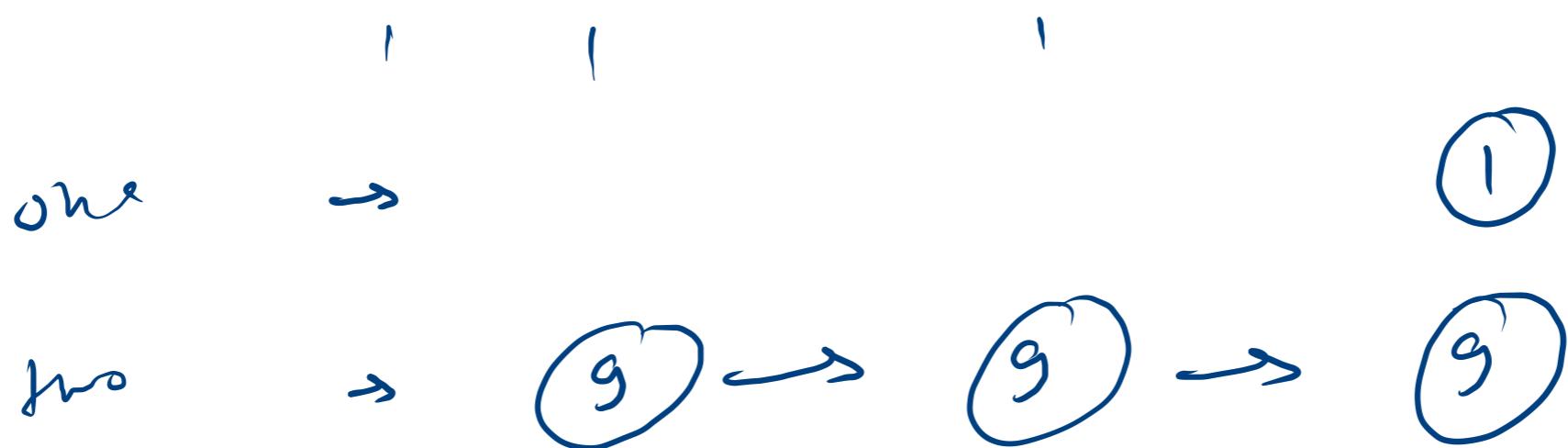
Data

1

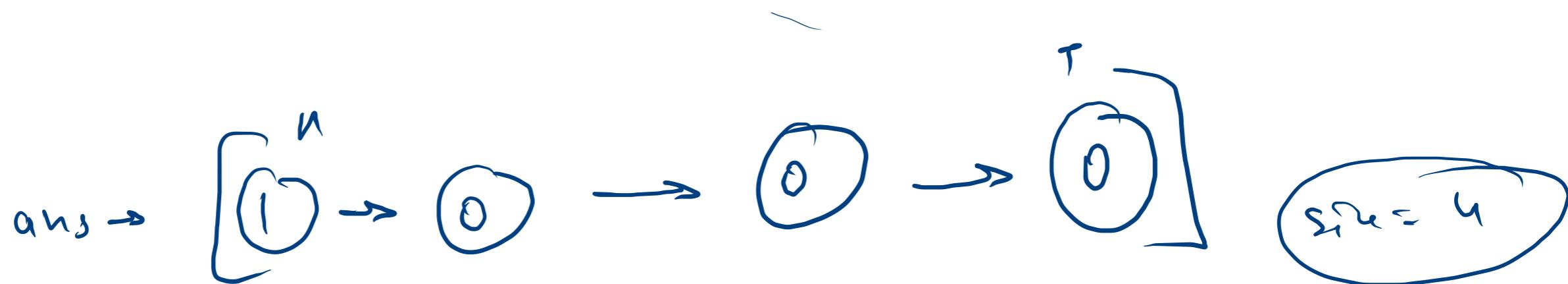
$$k = 3$$

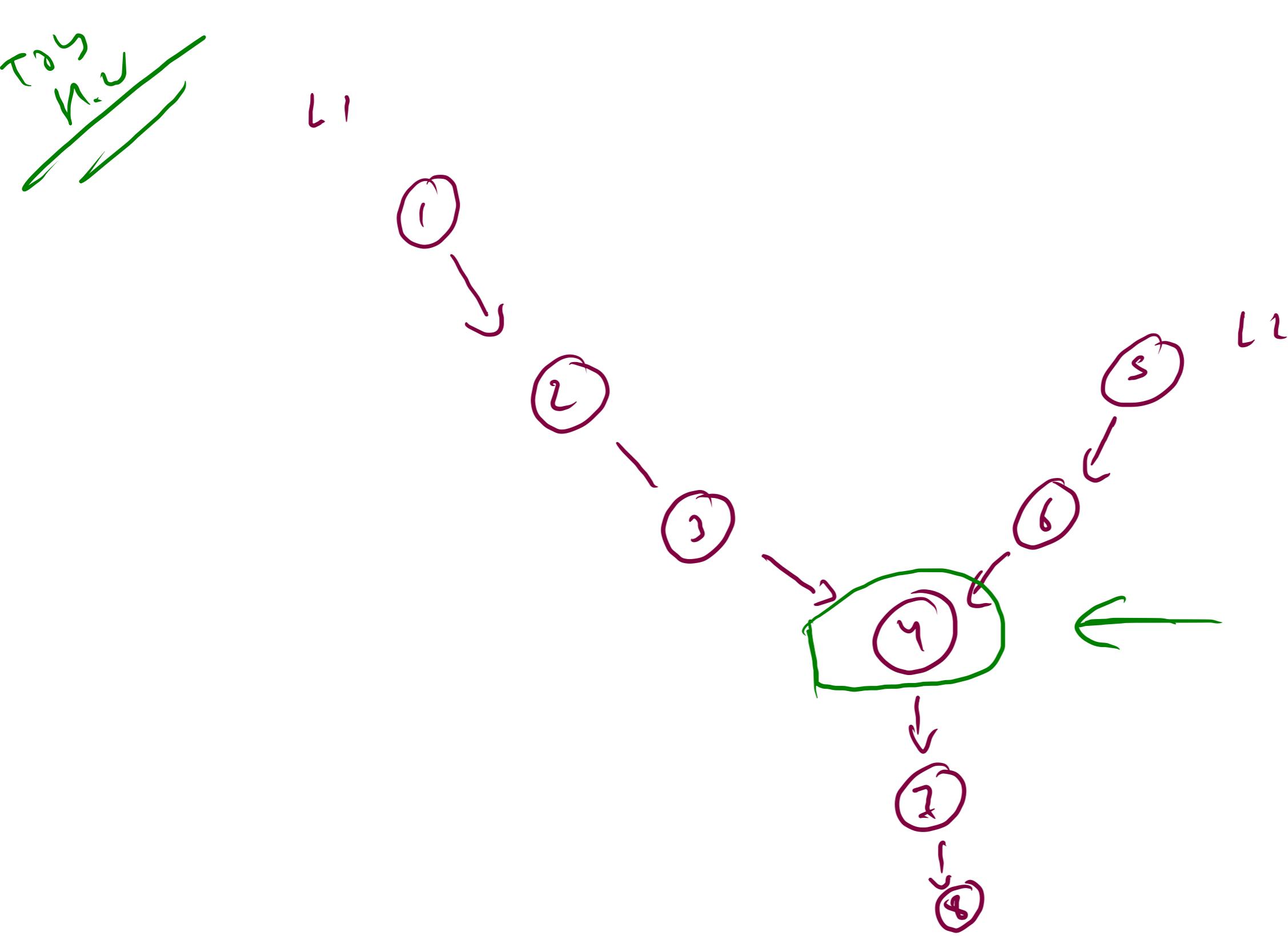
1

~~725 11. w~~



1
1
3
9 9 9
10
20





```

private void reversePRHelper(Node node){
    // write your code here
    if(node.next==null){
        head = node;
        return;
    }
    Node temp = node;
    reversePRHelper(node.next);
    node.next = temp;
    temp.next = null;
}

```

