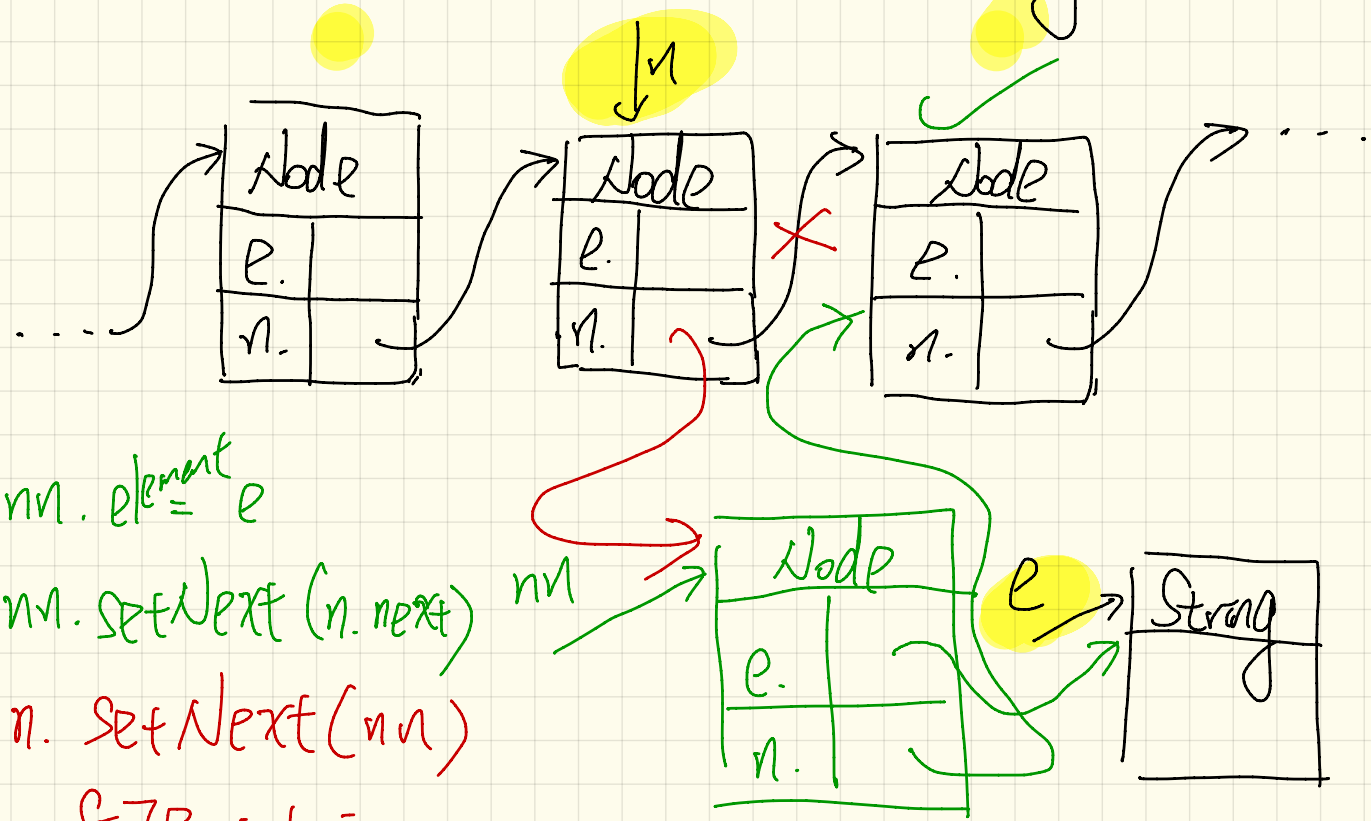
Lecture 15

Tuesday Oct. 31

void insertAfter(Node n, String e)



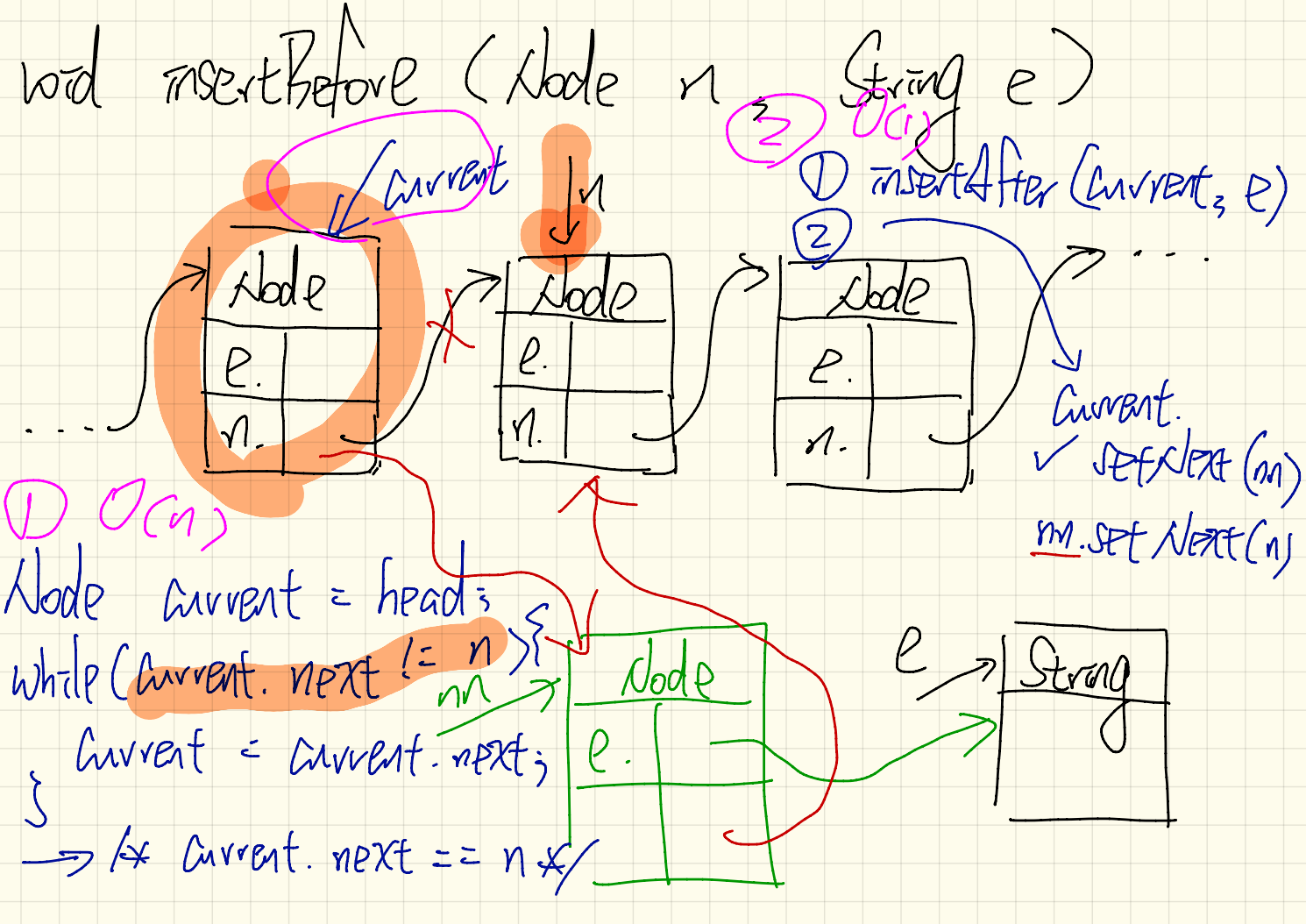
nn.element = e

nn.setNext(n.next)

n.setNext(nn)

size++;

void insertBefore (Node n, String e)



① O(n)

② O(1)

① insertAfter (current, e)

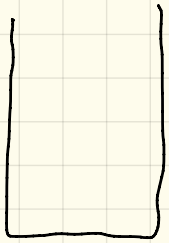
② current.
✓ setNext (nn)
nn.setNext(n)

```
Node current = head;
while (current.next != n){
    current = current.next;
}
→ /* current.next == n */
```

Stack Operations

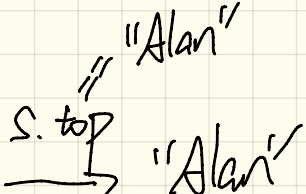
push, pop, top, size

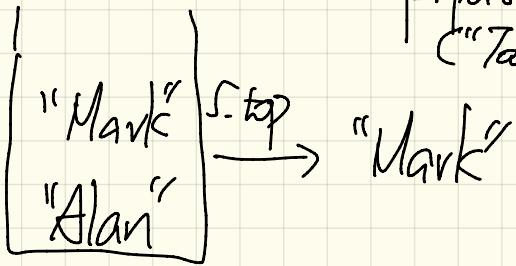
↳ ① return the current top
② remove the top

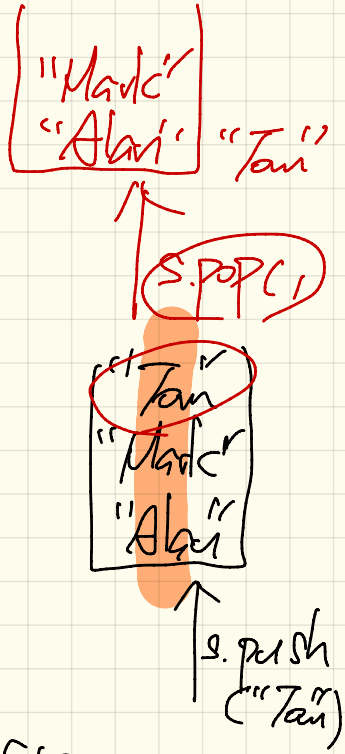


↓ S.push("Alan")

S.top → "Alan"







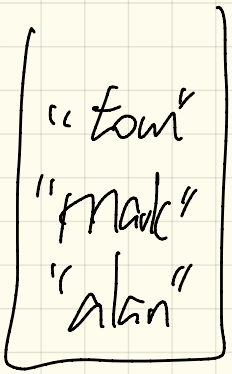
empty

Stack S

S.push("alan");

S.push("mark")

S.push("tom")



S.pop()

S.pop()

S.pop()

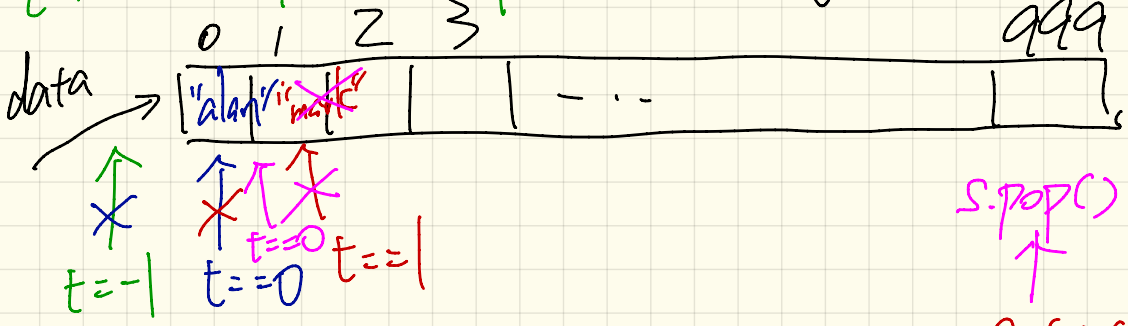
S.size() 0

"tom" "mark" "alan"

reverse order for pushing.

Implement a Stack using Array

t : index of the top.



s.pop() → data[t];
String ot = top();
data[t] = null;
t--;
return ot;

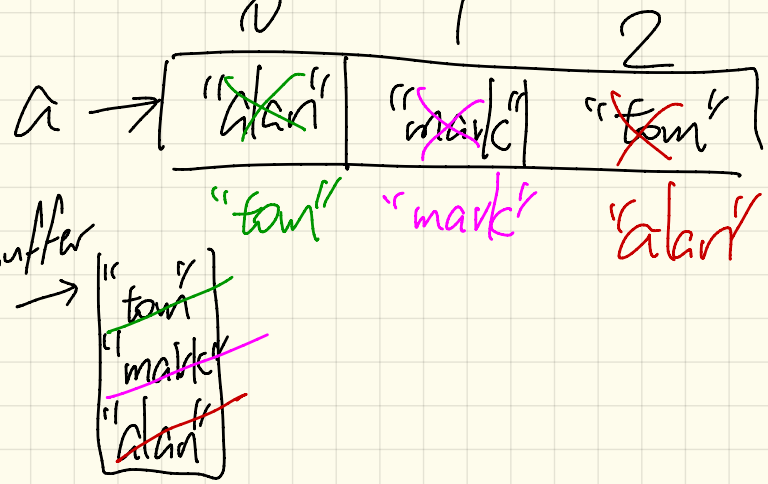
s.top() → data[t]

s.push("alan") → s.top() → s.push("mark")

↳ t++ ; data[t] = "alan"

↳ data[t]

↳ t++ ; data[t] = "mark"




```
int i=0;
while(!buffer.isEmpty()){
    a[i] = buffer.pop();
    i++;
}
```

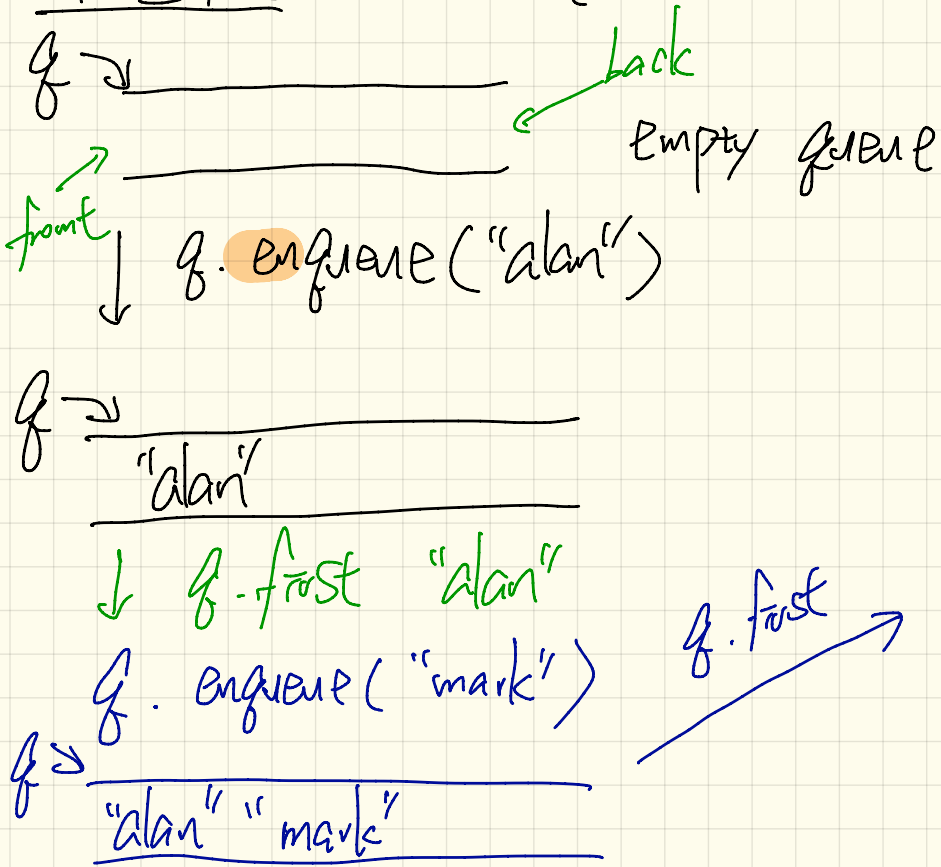
```
for(int i=0; i<a.length; i++){
    a[i] = buffer.pop();
}
```

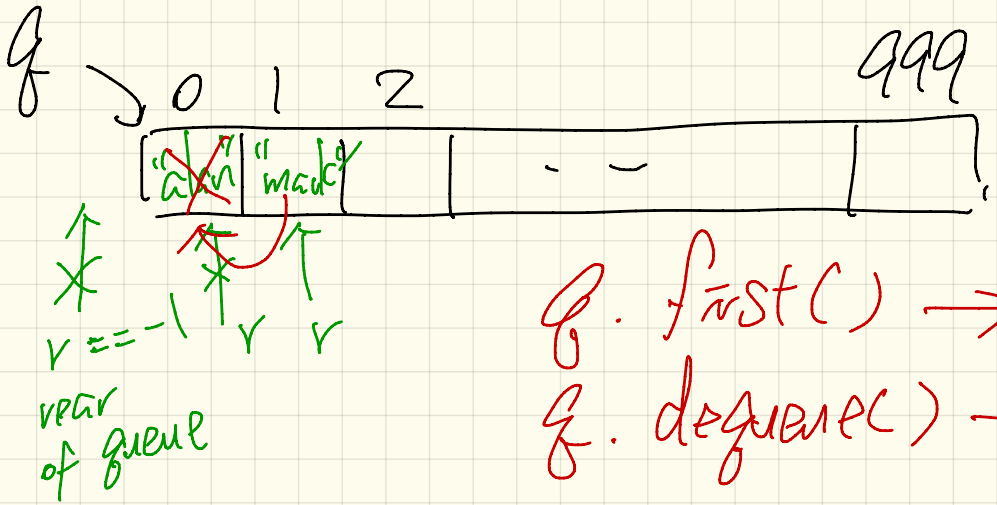
① i=0, "tom"

② i=1, "mark"

③ i=2, "alan"

FIFO Queue


q
back
empty queue
front
q.enqueue("alan")
q
"alan"
q.first "alan"
q.enqueue("mark")
q
"alan" "mark"
q.first
"alan"



q.first() → q[0]

q.dequeue() → O(n)

shift

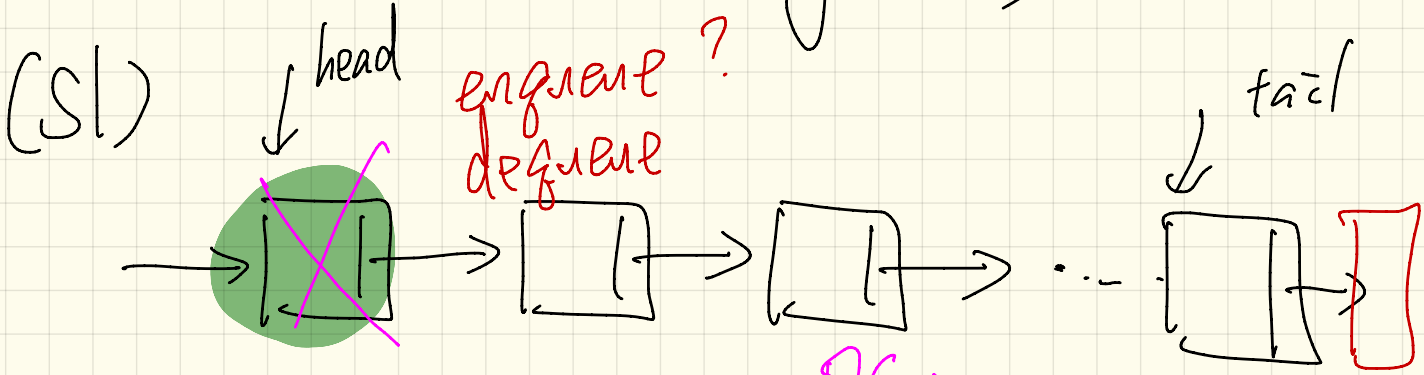
q.enqueue("alan") → q.enqueue("mavk")

↳ r++;
q[r] = "alan"

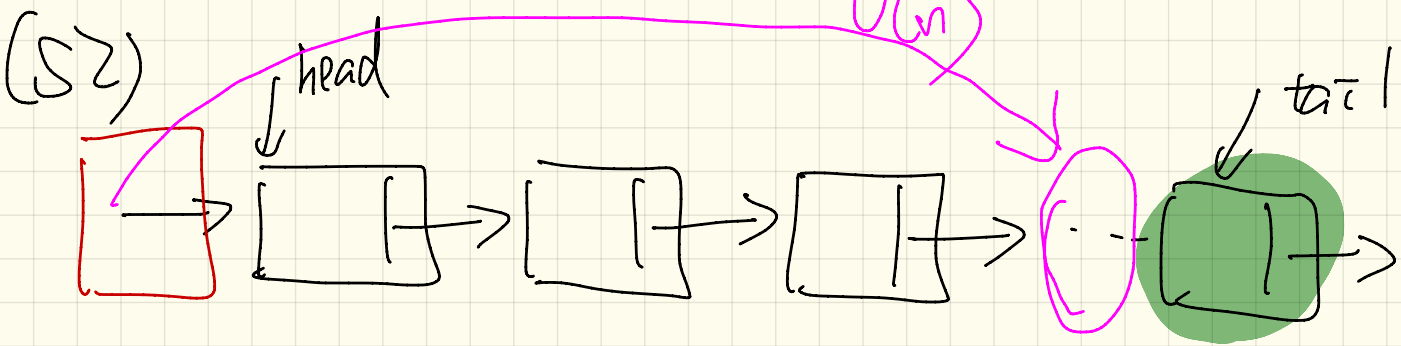
↳ r++;
q[r] = "mavk"

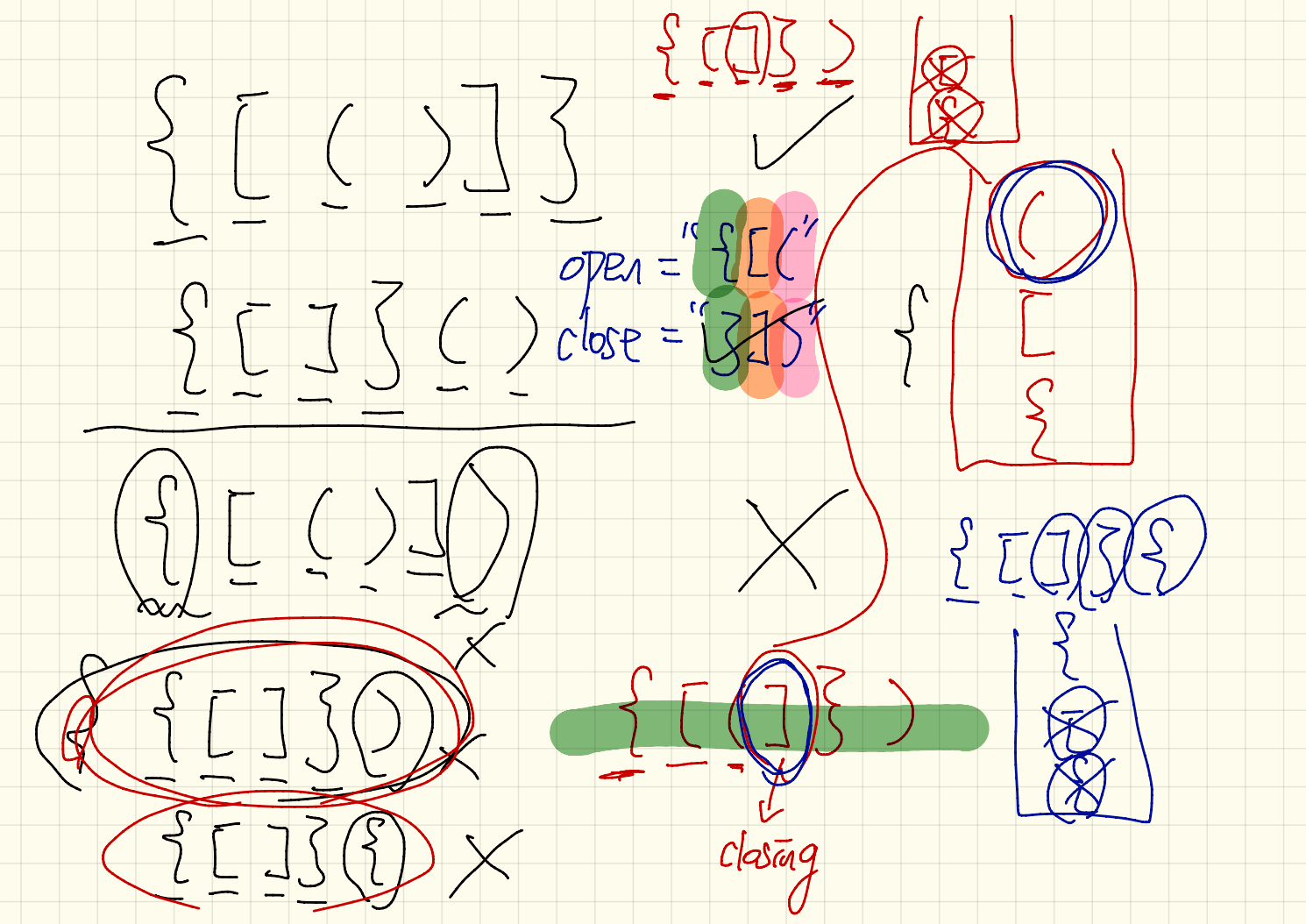
Implement the Q using SLL

(S1)



(S2)




open = "{[("
close = ")]}"
closing