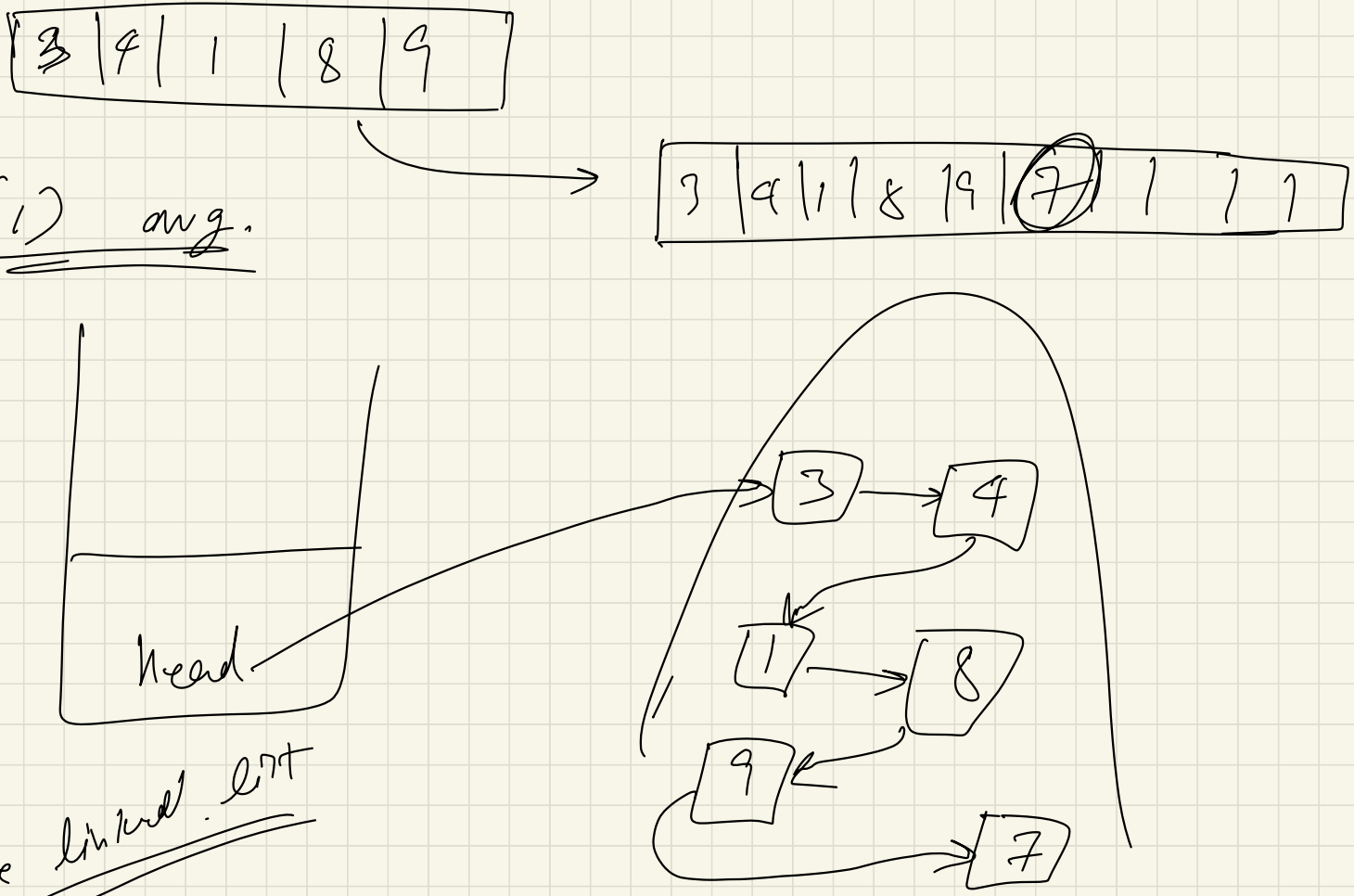
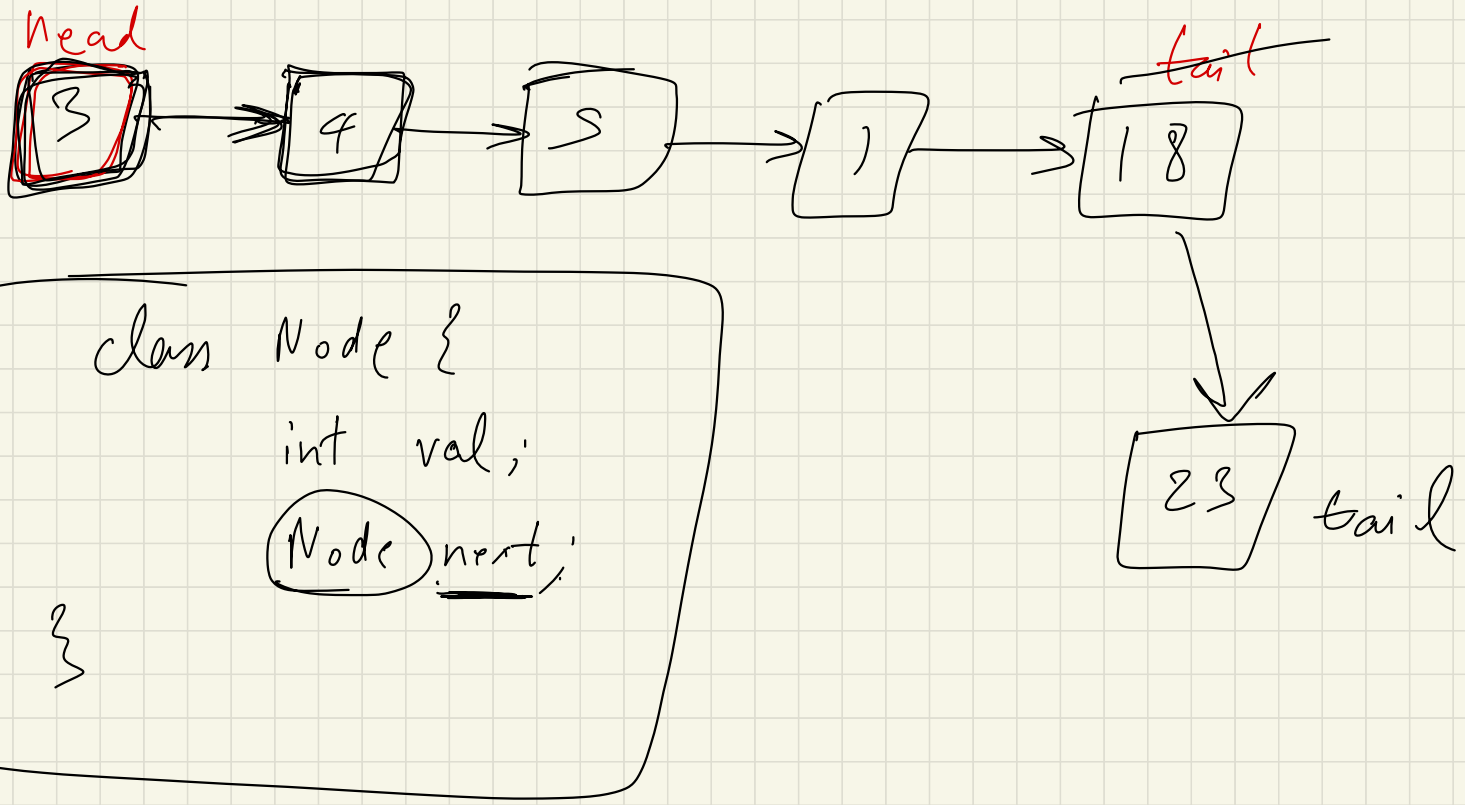
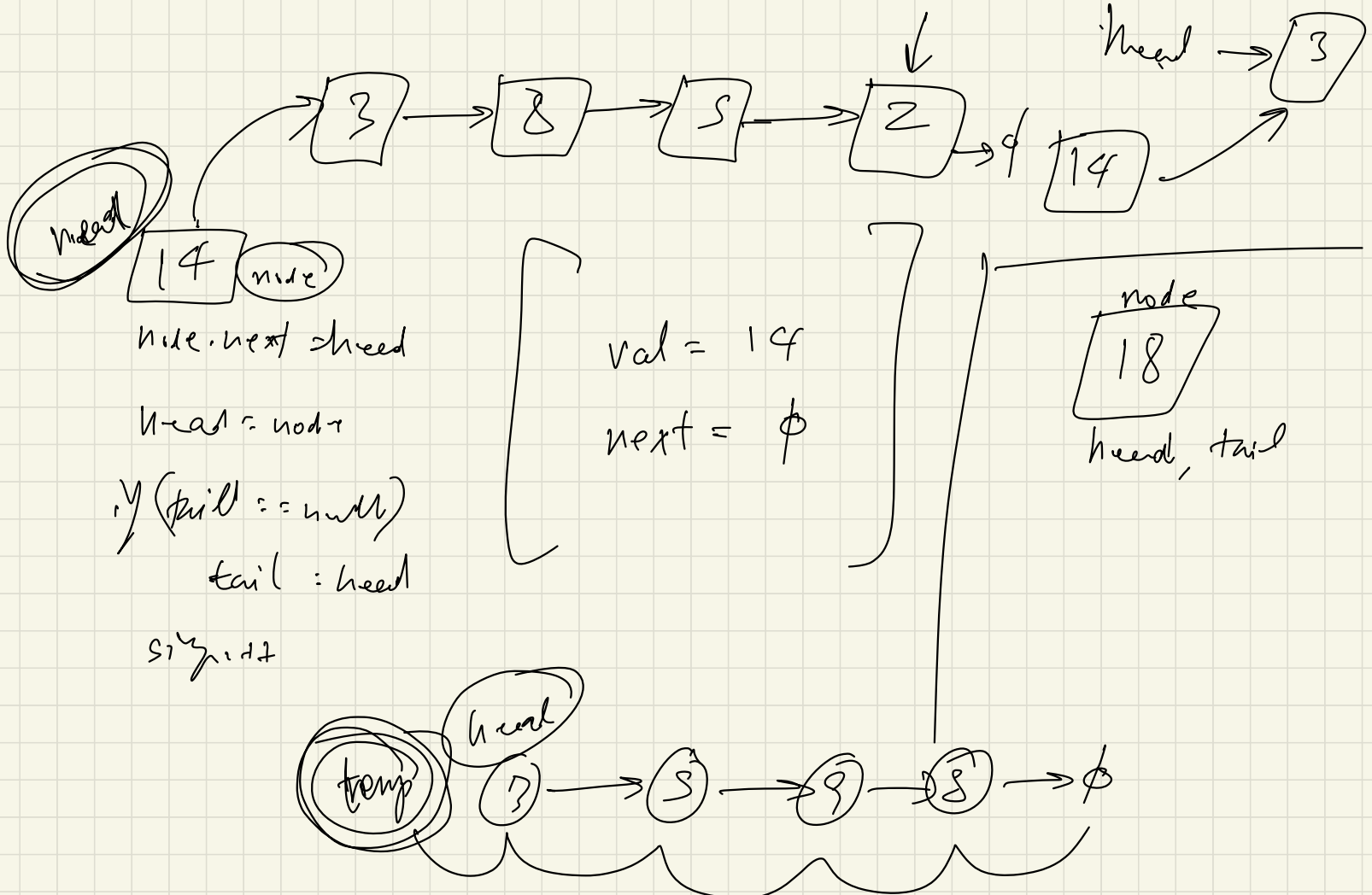
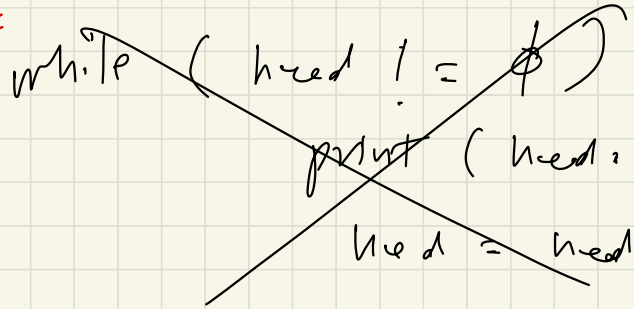
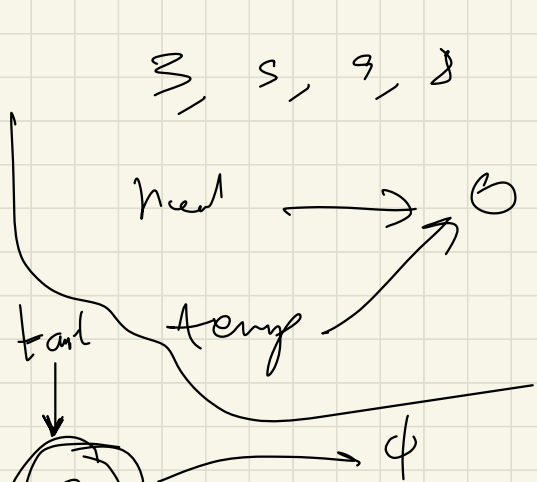

3 4 1 8 9
O(1) avg.
3 4 1 8 9 7
head
3
4
1
8
9
7
Single linked list

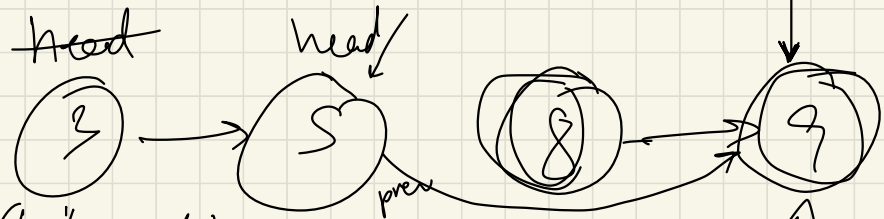

head
3
4
5
1
tail
18
23
tail
class Node {
int val;
Node next;
}


3
8
5
2
14
head
3
head
14
node
node.next = head
head = node
if (tail == null)
tail : head
size++
val = 14
next = ϕ
node
18
head, tail
temp
head
3
5
9
8
ϕ

Wrong





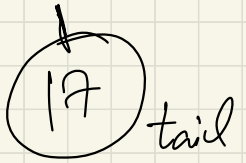


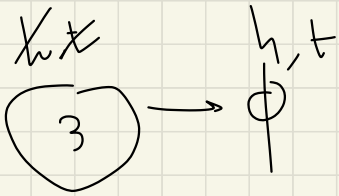
if (tail != null) { insertpost(val) };

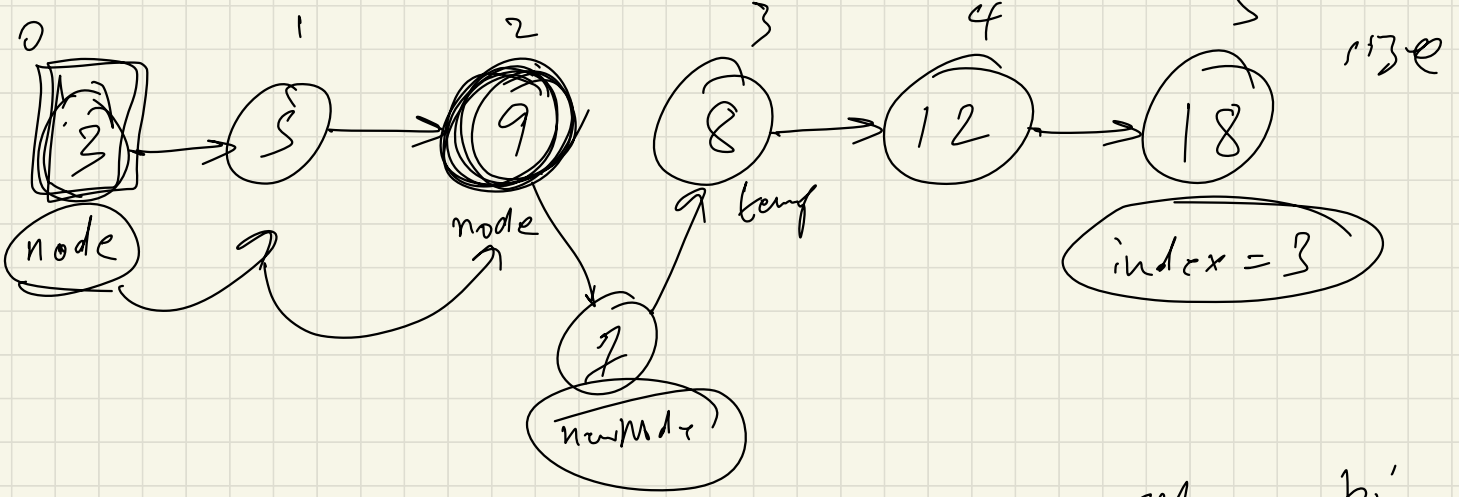
tail.next = node

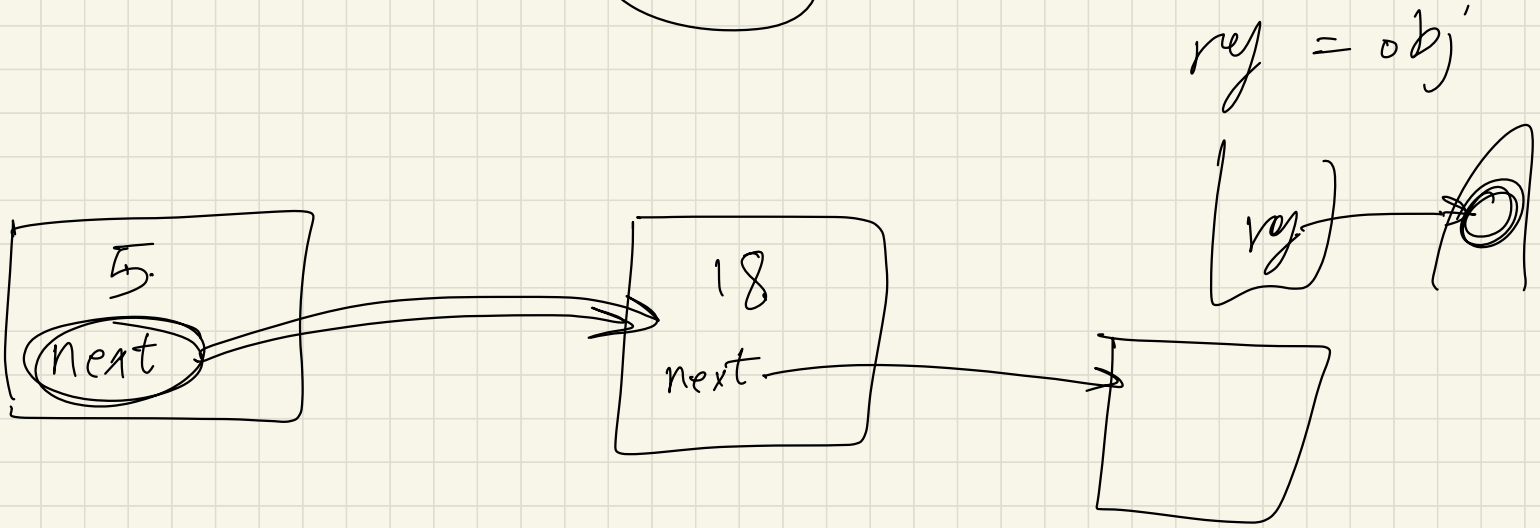
tail = node

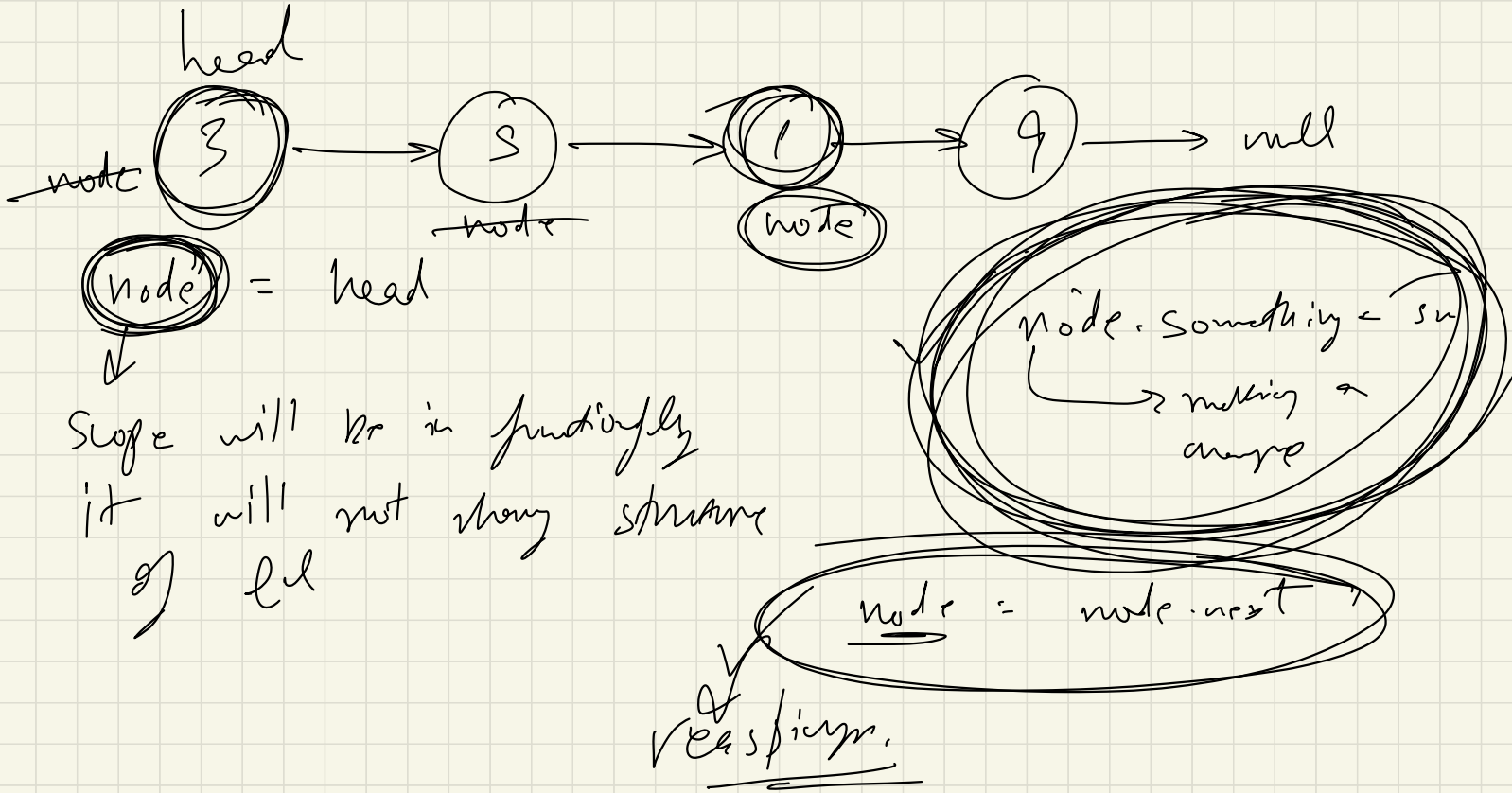
size++;



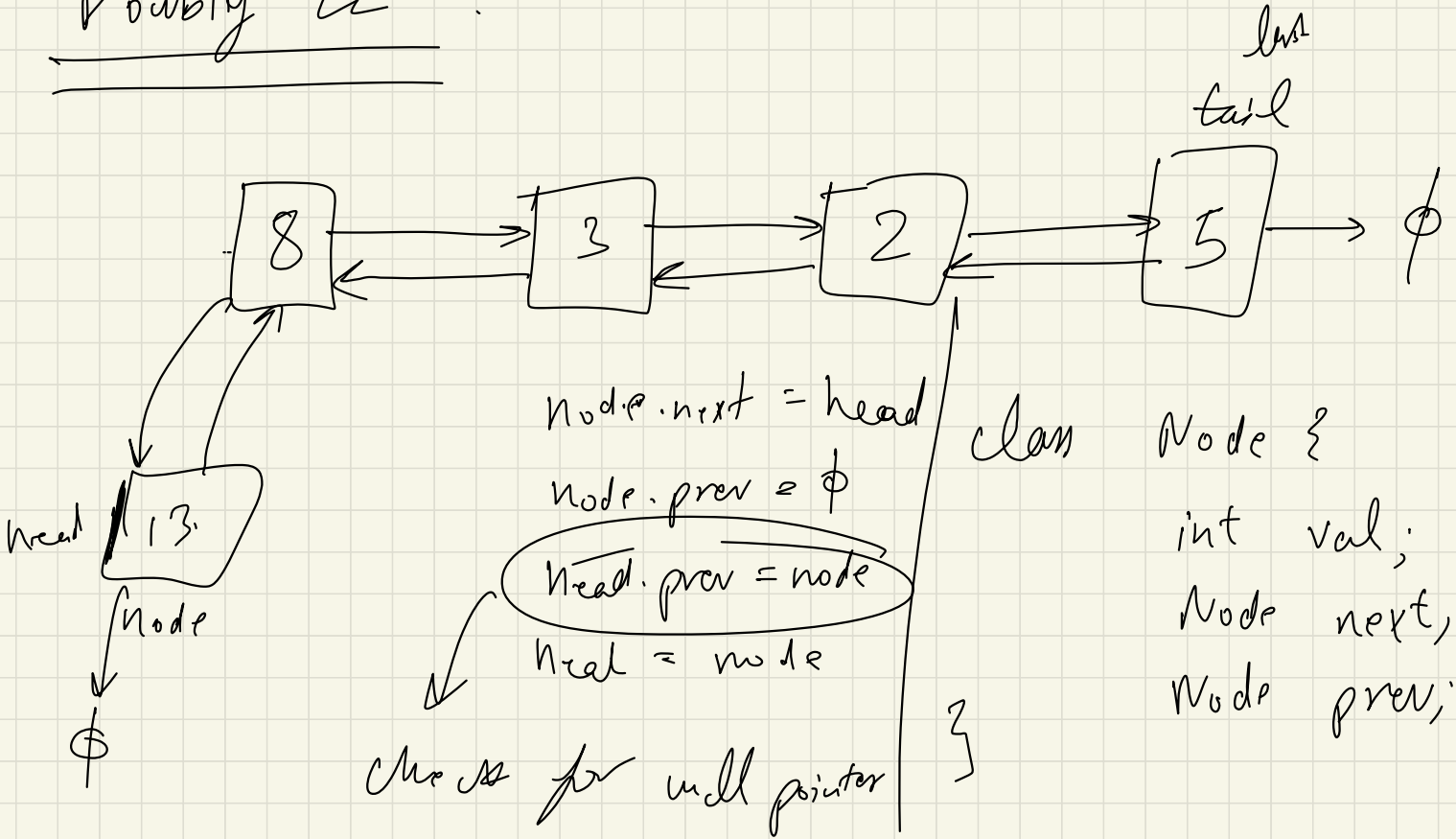


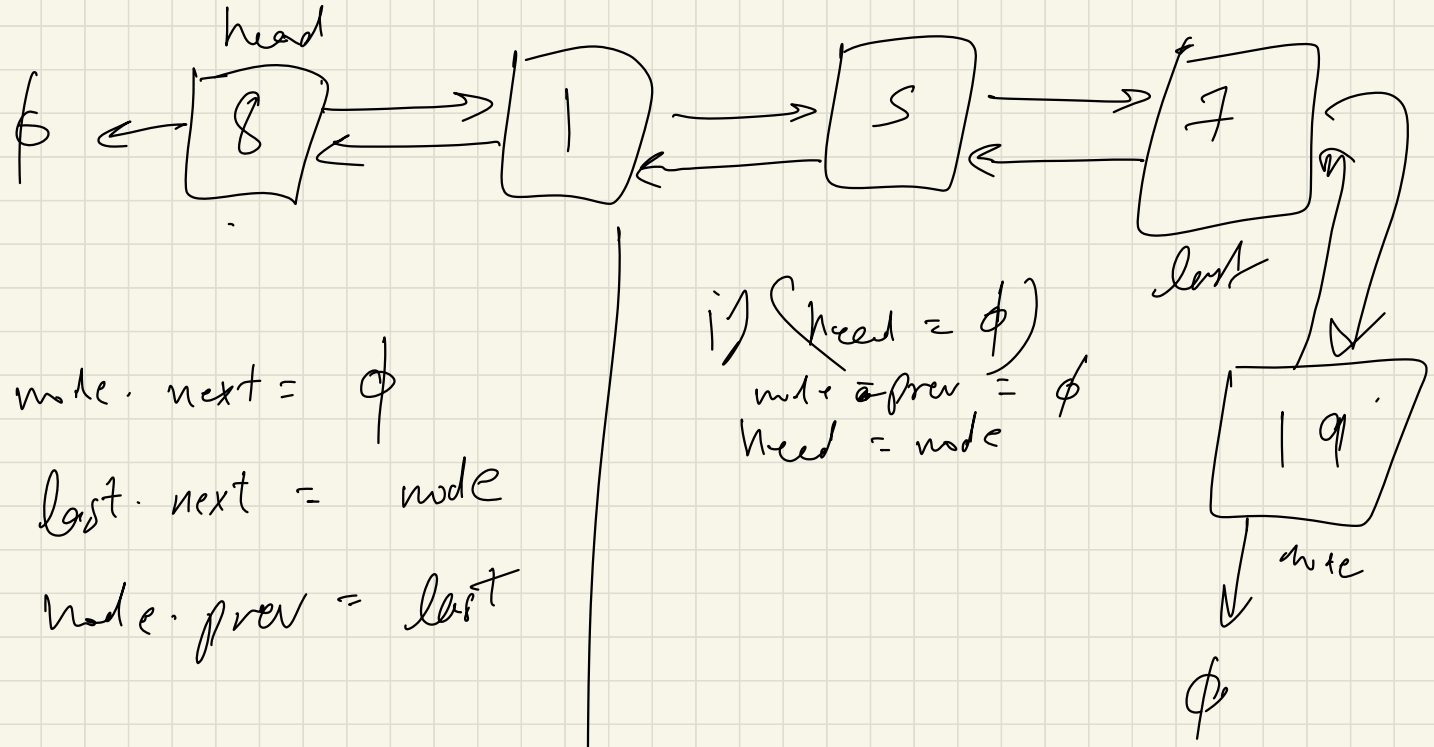

0
1
2
3
4
5
3
5
9
8
12
18
node
node
temp
index = 3
7
newNode

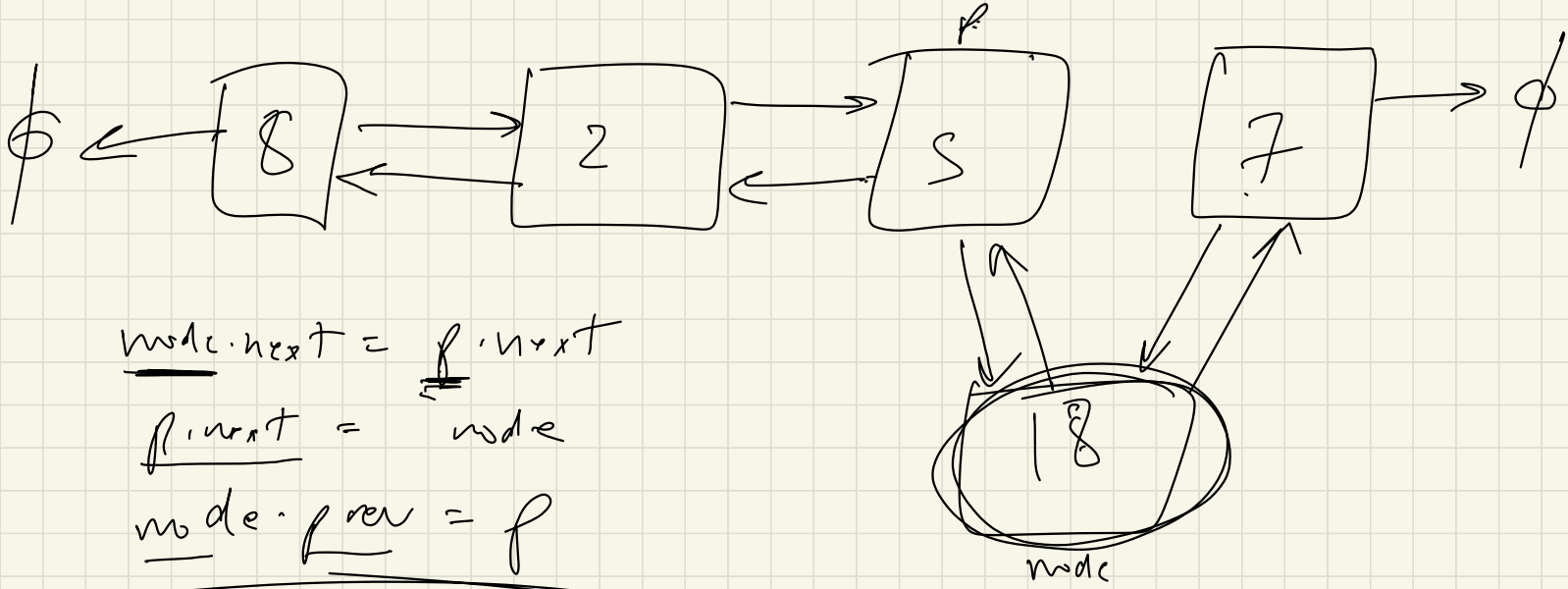

ref = obj
5
next
18
next
ref


head
3
5
1
9
null
node
node
node
node = head
Slope will be in functionality
it will not show structure
of ll
node.something = su
making a
change
node = node.next
reassigns.

Doubly LL :


tail
8
3
2
5
head
13
node
node.next = head
node.prev = ϕ
head.prev = node
head = node
Check for null pointer
class Node {
int val;
Node next;
Node prev;


head
ϕ
8
1
5
7
last
node
ϕ
node.next = ϕ
last.next = node
node.prev = last
if (head = ϕ)
node.prev = ϕ
head = node



node.next = p.next

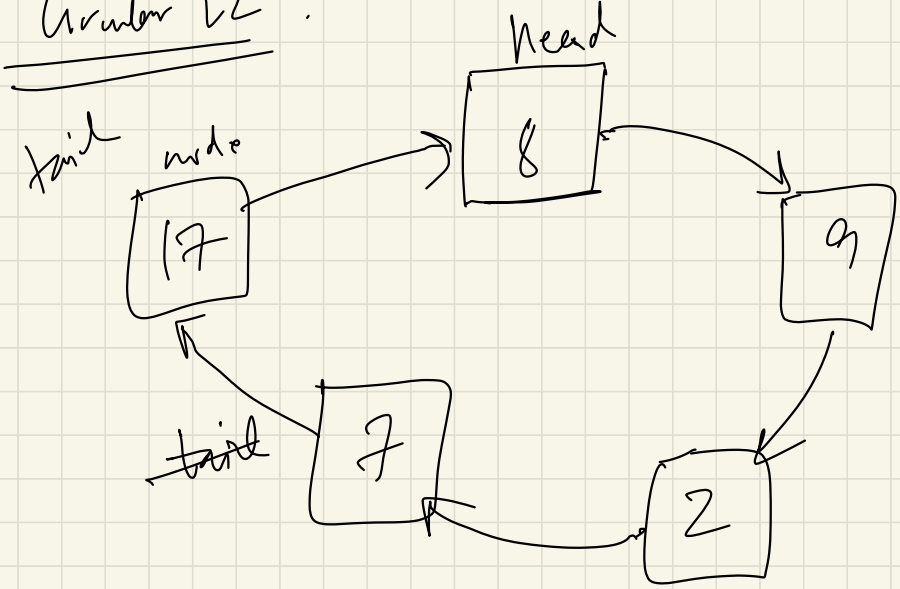
p.next = node

node.prev = p

node.next.prev = node

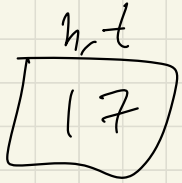
this may be null

Circular LL :



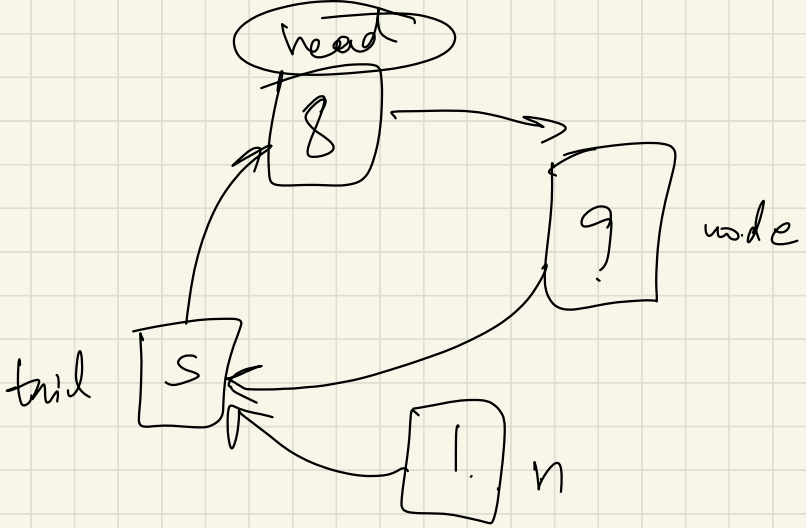

```
class Node{
    int val;
    Node next;
}
```

h, t




```
tail.next = node
node.next = head
tail = node
```

```
if (head == ϕ)
    head = node
    tail = node
```



val = 1