Linked List

22 March 2018 11:07

a b ~~c~~ d, e

↓

a b c d e

→ Insertion O(N) → Searching ↓ O(N)

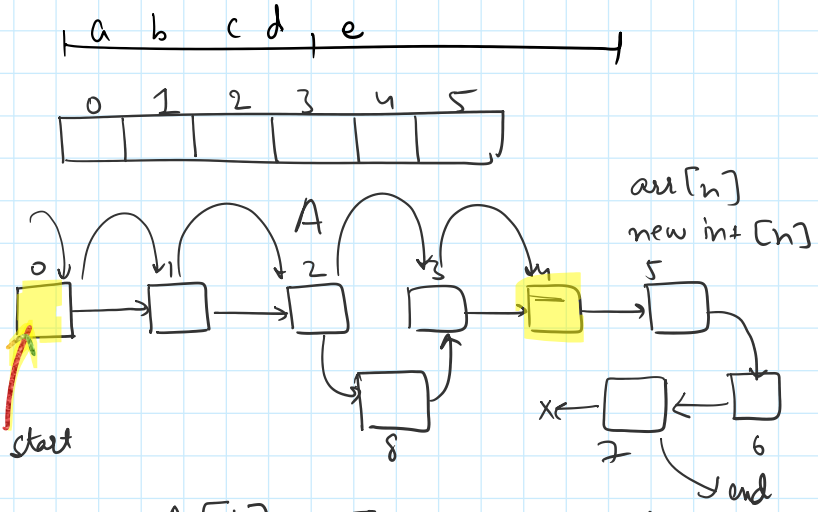
→ Deletion O(N)

→ Update O(N)

linked list

Problems with array: ① fixed size

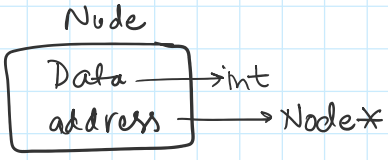
Advantage ② Access → O(1)

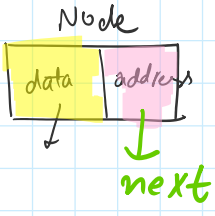


A[4] ✓ Traversing O(N)

Insertion O(1)

Deletion O(1)






```
class Node {
Public:
    int data;
    Node * next;

    Node () { data=0; next=NULL; }
    Node (int d) { data=d; next=NULL; }
};

Node n1;
```

n1

0	X

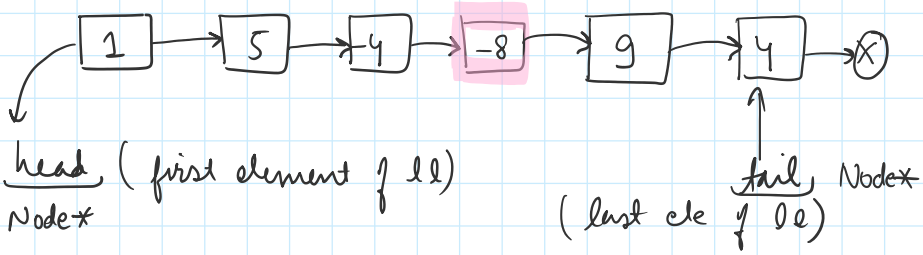
```
Node n2(3);
```

n2

3	X

Creating a list

1 5 -4 -8 9 4 -1

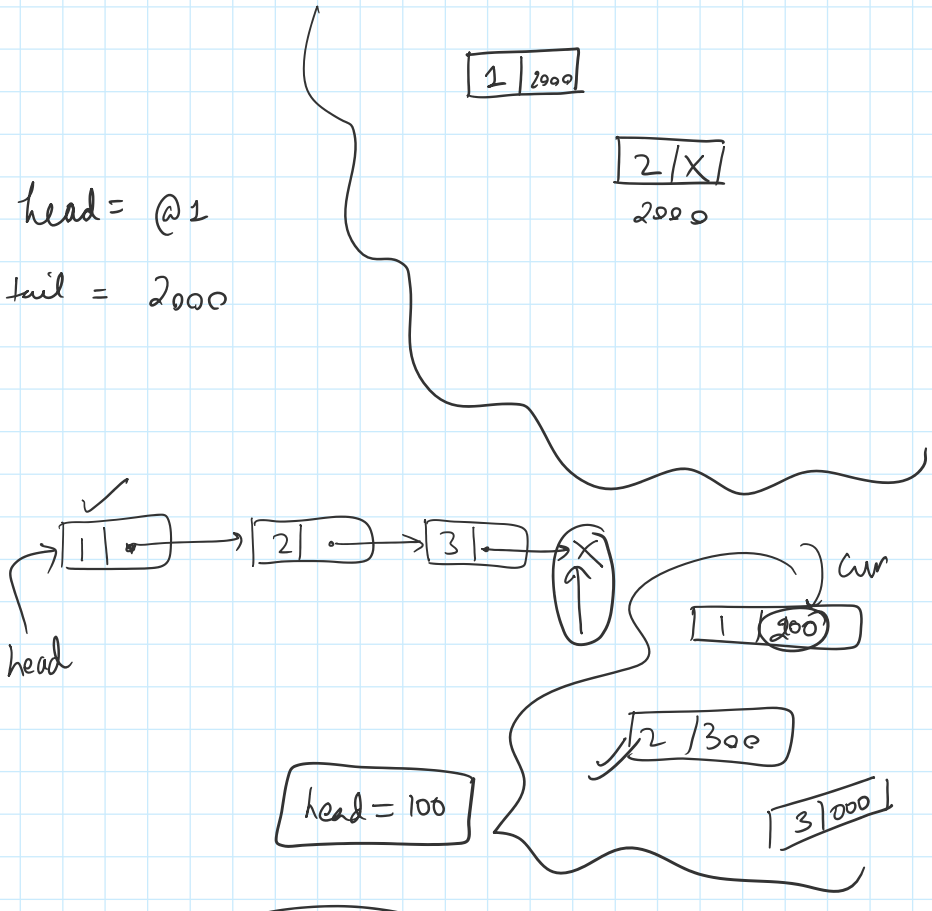


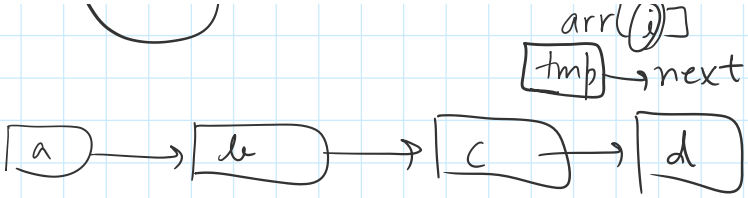
head = @1
tail = @1

(tail.next) = &5

(*tail).next ≡ tail->next; ✓
node

n.data Node *n;
head->next; Node *head = &n;



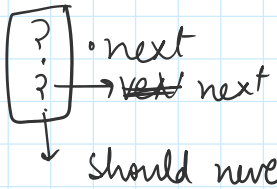


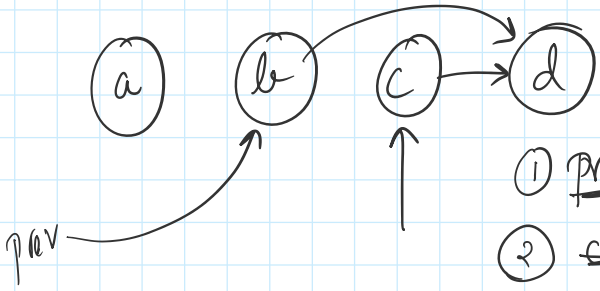
tail → next

(*tail)

(*NULL)

(*Garbage) X

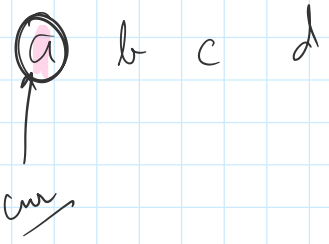


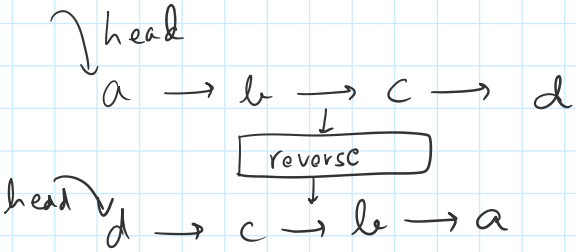


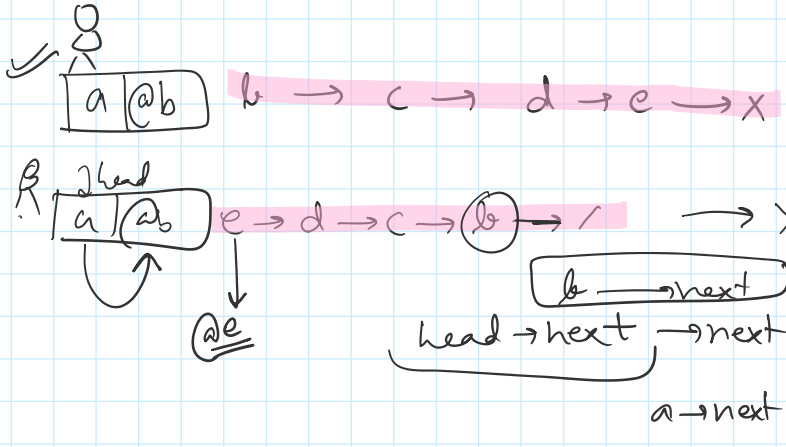
a

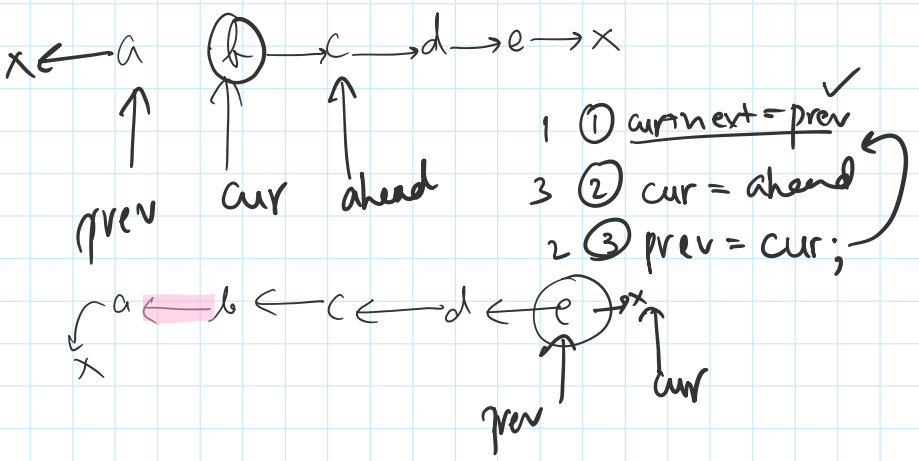
① prev → next = cur → next

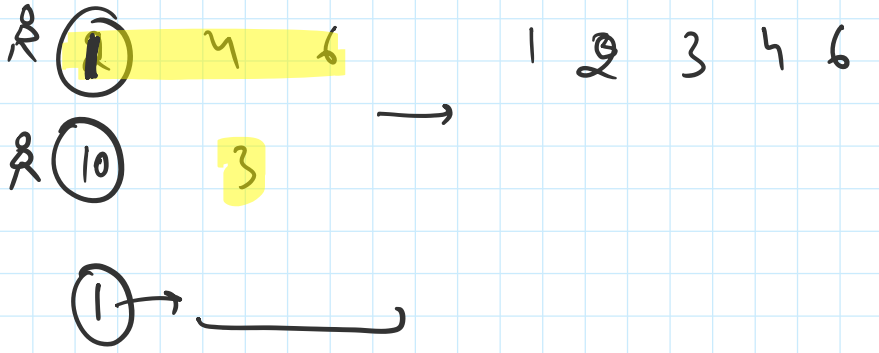
② ~~cur~~ delete cur;

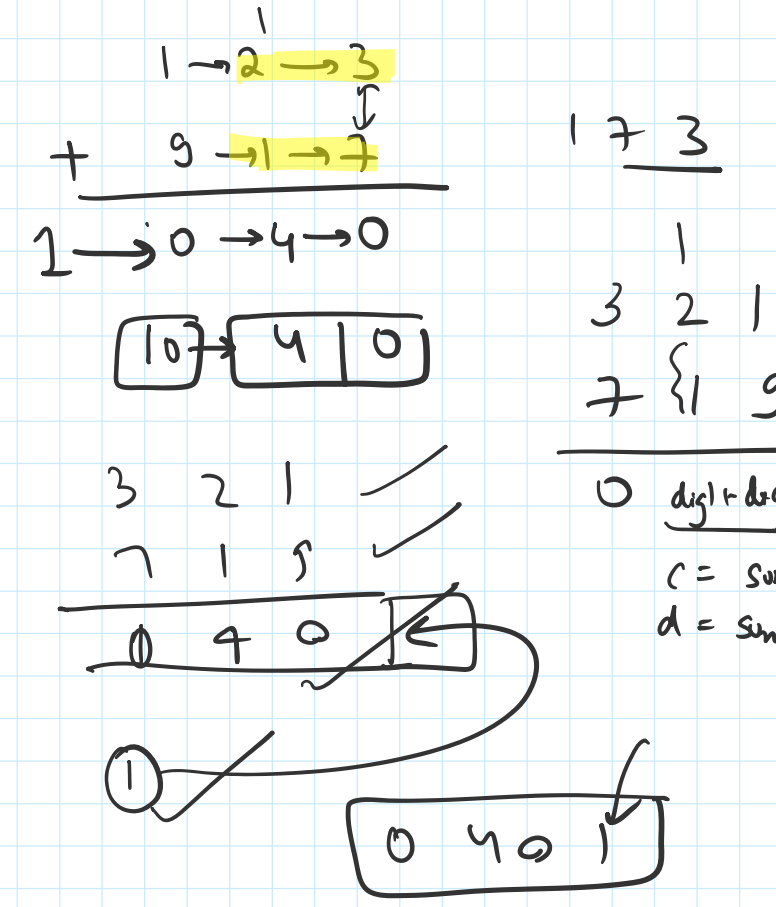


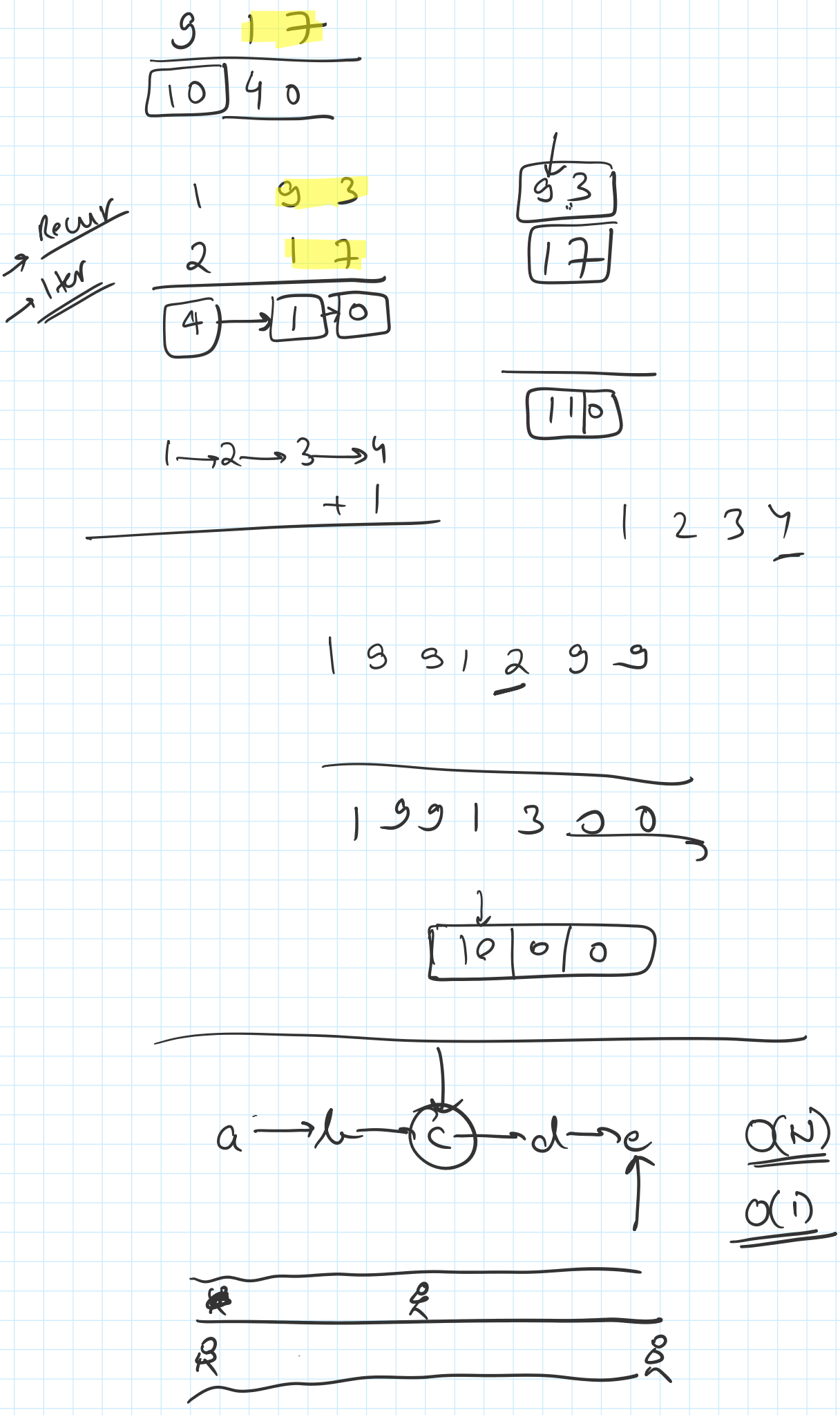


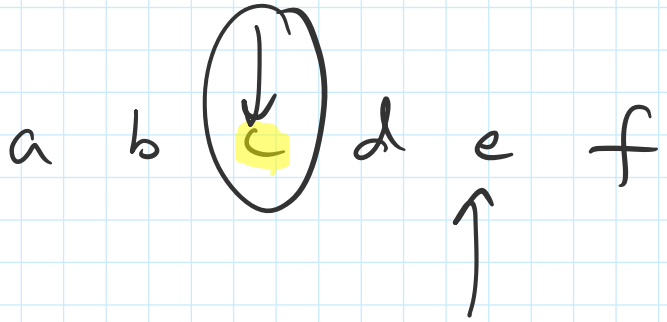



prev
cur
ahead
1 ① curr.next = prev
3 ② cur = ahead
2 ③ prev = cur;
prev
cur

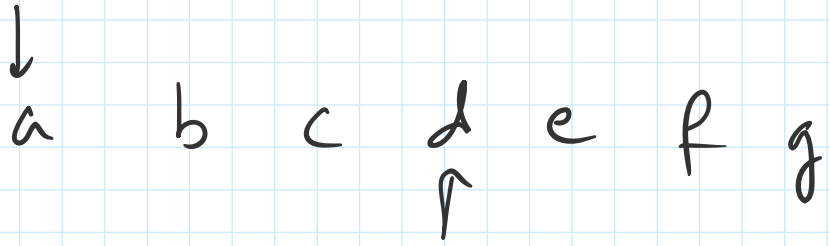

1 2 3 4 6
10
3


1 → 2 → 3
+ 9 → 1 → 7
1 → 0 → 4 → 0
10 → 4 | 0
1 7 3
1
3 2 1
7 1 9
0 digit = d1+d2+c
c = sum/10;
d = sum%10
3 2 1
7 1 9
0 4 0
1
0 4 0 1
1 2 3


9 1 7
10 4 0
Recur
Iter
1 9 3
2 1 7
4 → 1 → 0
9 3
1 7
1 1 0
1 → 2 → 3 → 4
+ 1
1 2 3 4
1 9 3 1 2 9 9
1 9 9 1 3 0 0
1 0 0 0
a → b → c → d → e
O(N)
O(1)

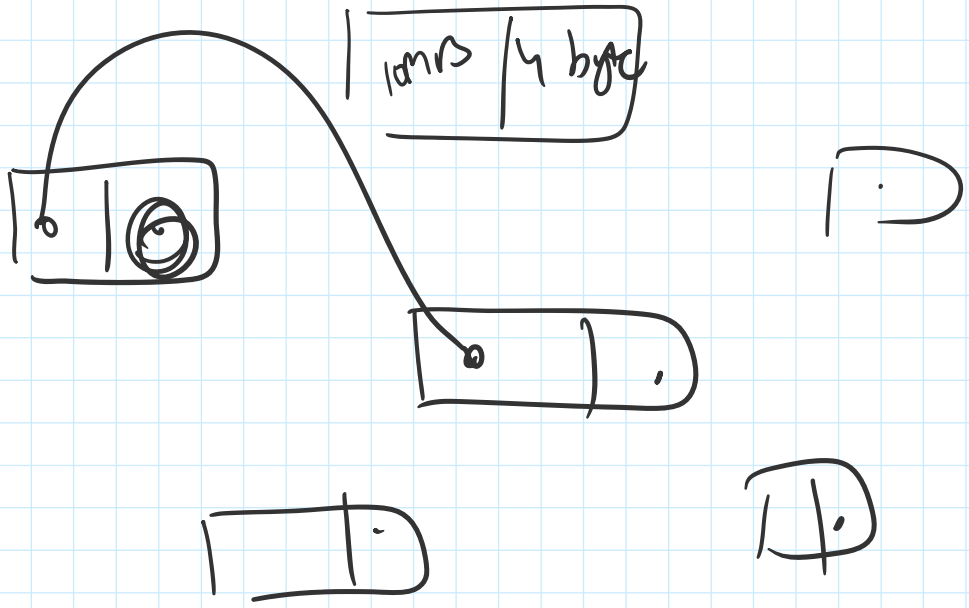


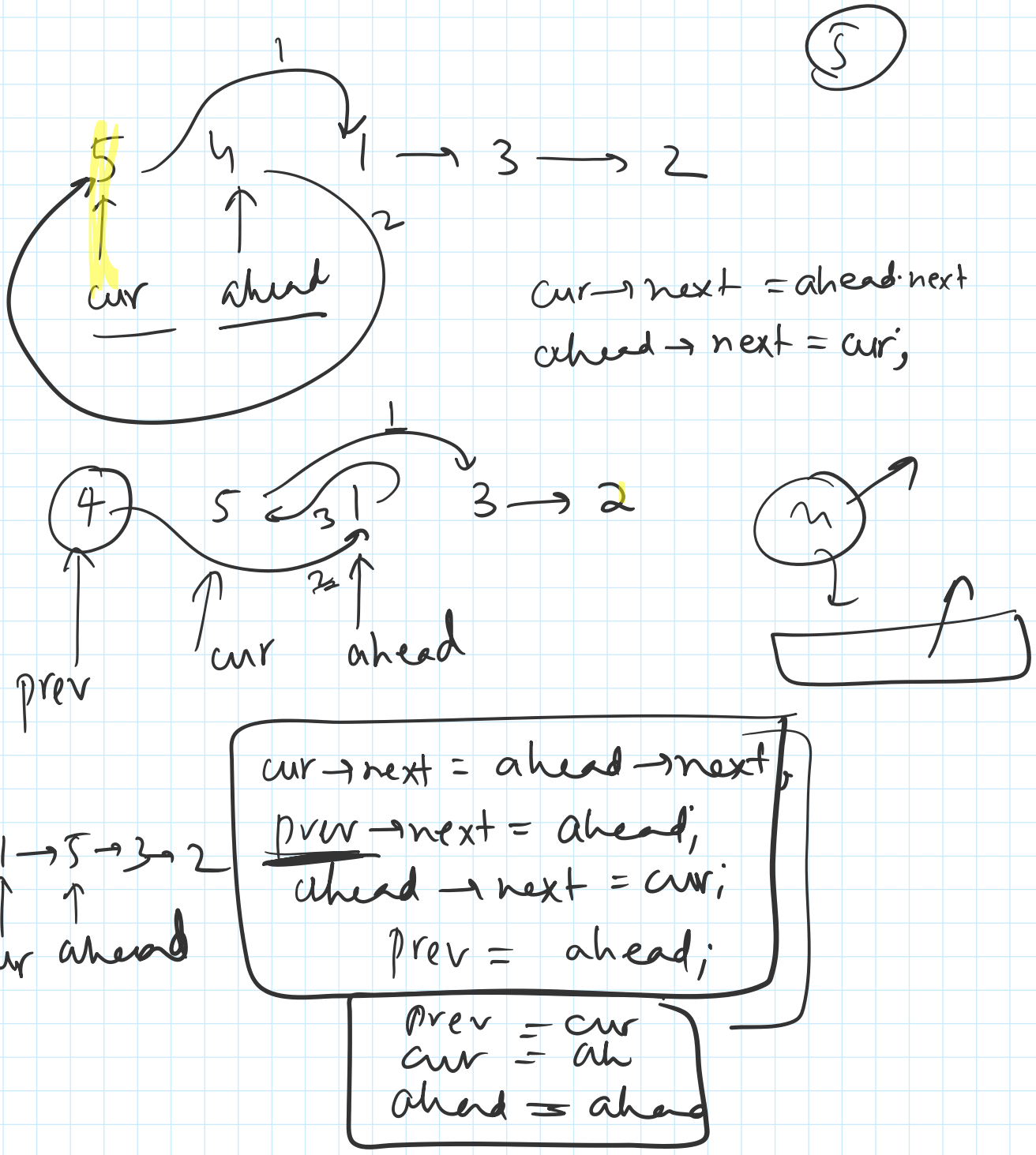
5th node from last.



Bubble Sort

5 4 1 3 2




5
1
5
4
1 → 3 → 2
2
cur
ahead
cur→next = ahead·next
ahead→next = cur;
1
4
5
3
1
3 → 2
2
prev
cur
ahead
n
f
cur→next = ahead→next;
prev→next = ahead;
ahead→next = cur;
prev = ahead;
4→1→5→3→2
cur
ahead
prev = cur
cur = ah
ahead = ahead
5
4
prev
cur
if (swap) {
if (head)
if else
}
2:30

else ~~do~~ {

3

1 2345

---

a (b c d) → n1

(e f g) → n2