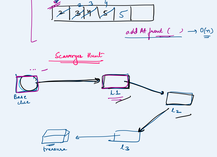
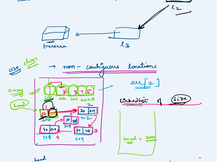
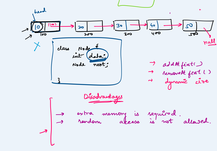
Thursday, 11 April 2021 8:34 PM

Linked List







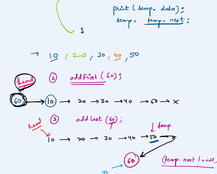
Disadvantages

→ extra memory is required.
→ random access is not allowed.

---

operations on Linked List

① Traverse the entire linked list.



→ 10 ; 20, 30, 40, 50

```
Node temp = head;
while ( temp != null) {
    print (temp.data);
    temp = temp.next;
}
```

② add First (60)

③ add at front (60)

④ add At (60, 2)

⑤ Remove First()

⑥ remove last

⑦ remove At (int idx)

```
Node temp2 = temp.next;
temp.next = temp.next.next;
temp2 = null;
```

Q1 Reverse a linked list.

Q2 Given a linked list, determine if

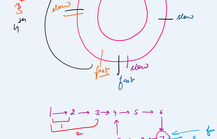
(a) there is a cycle in it.

(b) detect the starting point of cycle

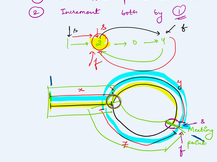
① Approach ①

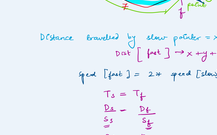
```
class Node {
    int data;
    Node next;
    boolean visited;
}
```

② Floyd cycle finding algorithm



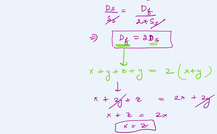
2 * speed (slow) = speed (fast)

① 

② 

① Move slow pointer to head

② Increment both by 1



Distance travelled by slow pointer = x + y

Dist [fast] → x + y + z + y

speed [fast] = 2 * speed [slow]

$T_s = T_f$

$\frac{D_s}{S_s} = \frac{D_f}{S_f}$

$\frac{D_f}{D_s} = \frac{S_f}{S_s}$

$\frac{D_f}{D_s} = \frac{D_f}{2 * S_s}$

$\Rightarrow \boxed{D_f = 2D_s}$

$x + y + z + y = 2(x + y)$

$x + \cancel{y} + z = 2x + \cancel{2y}$

$x + z = 2x$

$\boxed{x = z}$