

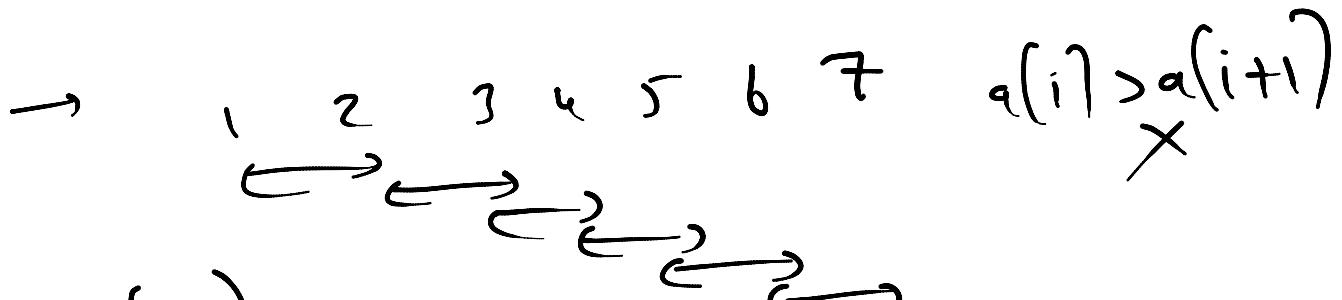
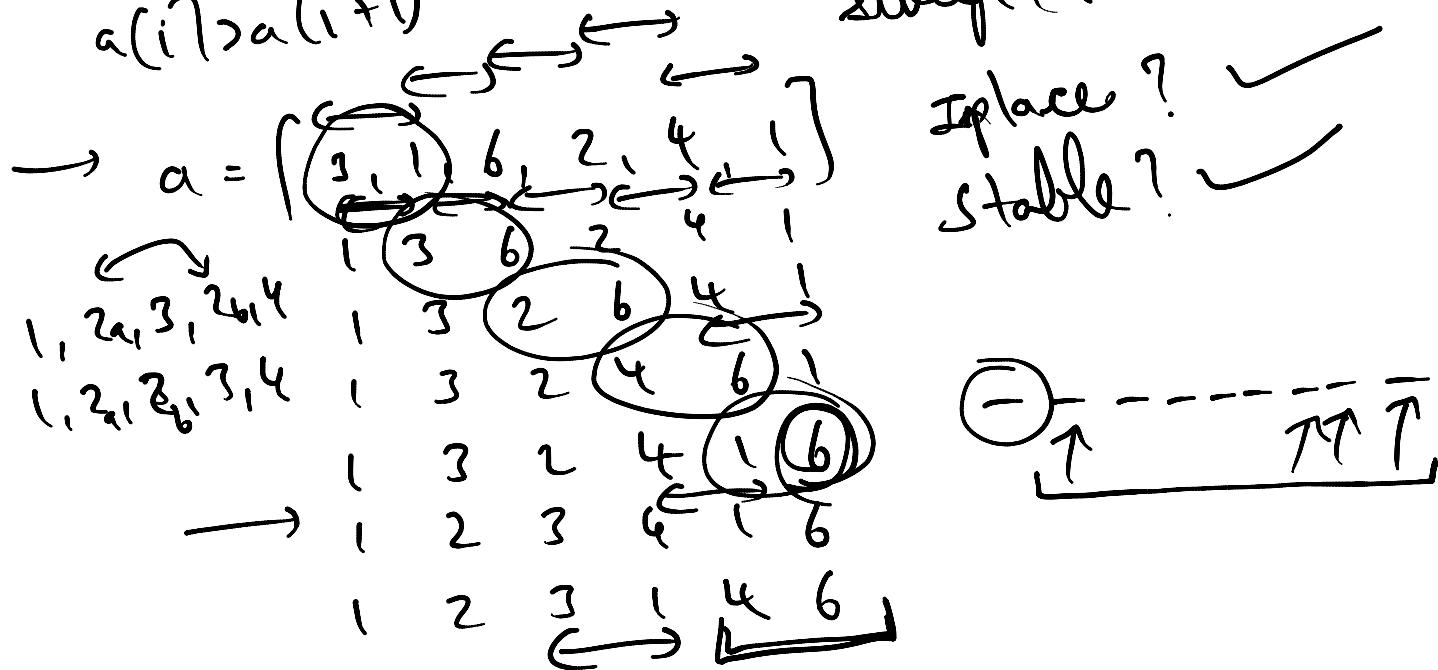
$\Rightarrow$  Bubble Sort :-

$a(i) > a(i+1)$

$i, i+1$

$a(i) > a(i+1)$

swap  $(a(i), a(i+1))$



$$TC = O(n^2)$$

$$AS = O(1)$$

$\Rightarrow$  Selection Sort :-

Select min/max element at every iteration and place it into its correct position.

correct position.

$$a = \{3, 8, 1, 5, 7, 2, 4\}$$

$$\begin{array}{c} \uparrow \\ i=0 \\ a = \{ \boxed{1, 8, 3, 5, 7, 2, 4} \} \\ \uparrow \\ 1, 2, 3, 5, 7, 8, 4 \\ \uparrow \\ 1, 2, 3, 4, 7, 8, 5 \\ \uparrow \\ 1, 2, 3, 4, 5, 8, 7 \\ \uparrow \\ 1, 2, 3, 4, 5, 7, 8 \end{array}$$

TC =  $O(n^2)$

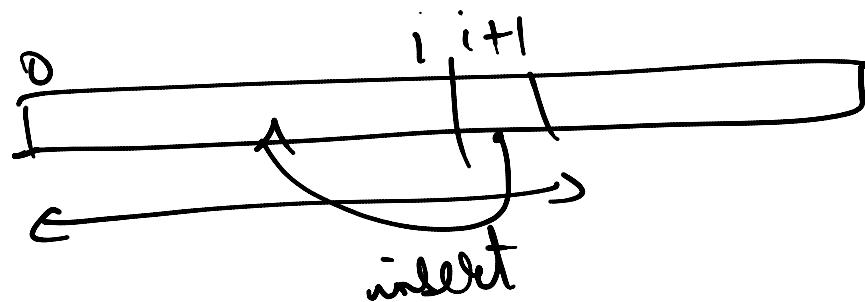
AS =  $O(1)$

Inplace ✓  
stable X

$$\begin{array}{l} a = \{ \boxed{5_a, 5_b, 1} \} \\ a = \{ \boxed{1, 5_b, 5_a} \} \end{array}$$

⇒ Insertion Sort :-

→  $i \rightarrow$  prefix  $(0:i)$  sorted  
 $a(i+1) \rightarrow$  prefix  $(0:i+1)$  sorted



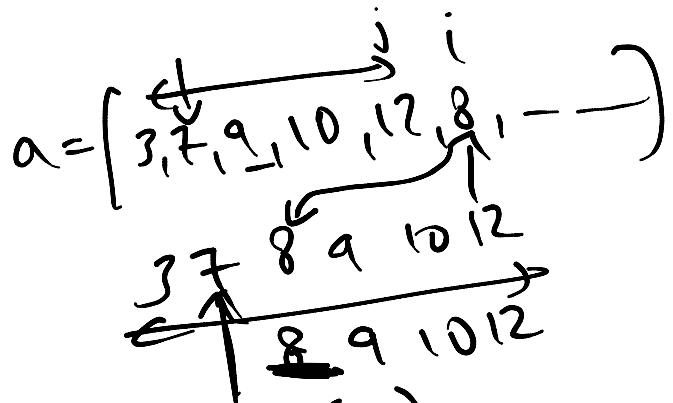
$$a = [3, 2, 1, 5, 4]$$

$$a = [2, 3, 1, 5, 4]$$

$$a = [1, 2, 3, 5, 4]$$

$$a = [1, 2, 3, 4, 5]$$

insert



$$TC = O(n^2)$$

$$AS = O(1)$$

Inplace ✓  
stable ✓  
 $2a \leftarrow 2b$

Floyd Warshall Algo

$\tau$

↓ + shortest path algo

$\tau_{G_2}$

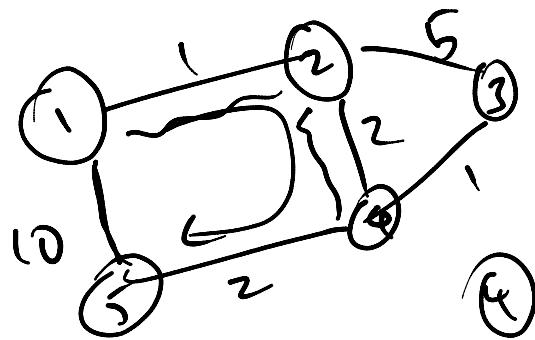
Used to find shortest path b/w every pair of vertices.

→ Find shortest path b/w 2 vertices with 1 edge

⑥ →

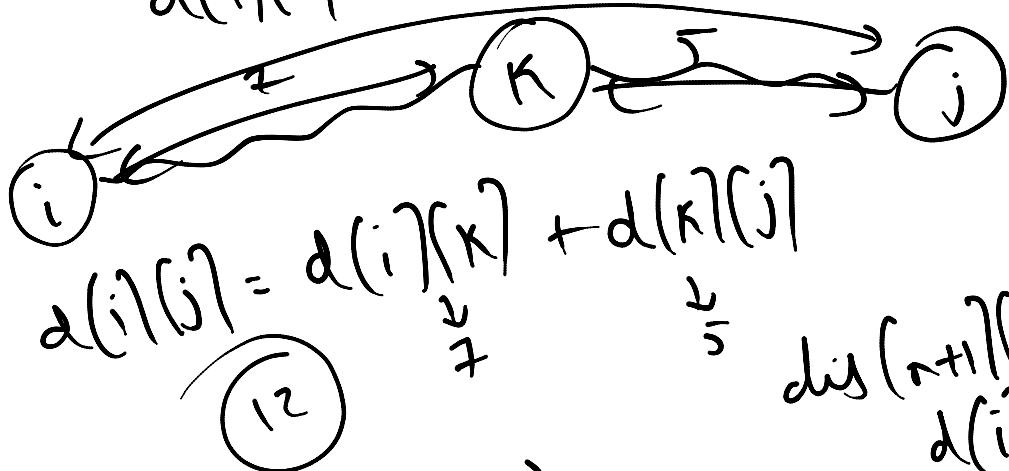
→ 1 → 2 → 5 →

2 edges  
3 edges  
4 edges



6 edges  
 $\vdots$   
 $(n-1)$  edges.

$$d(i, j) = d(i, k) + d(k, j)$$



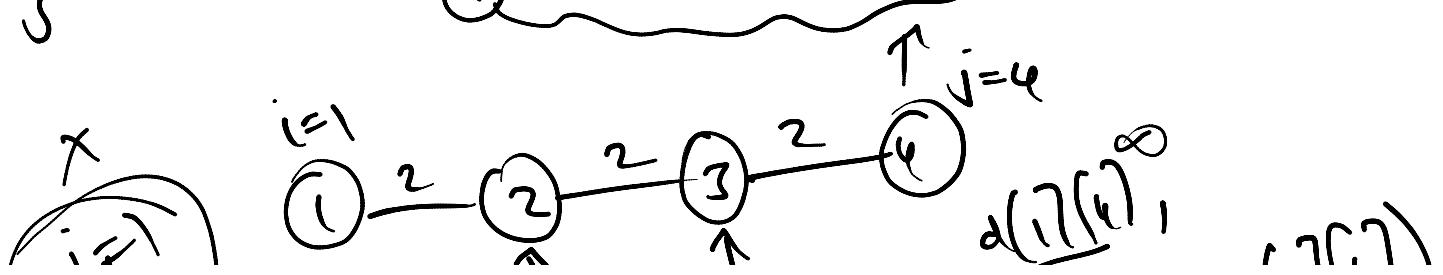
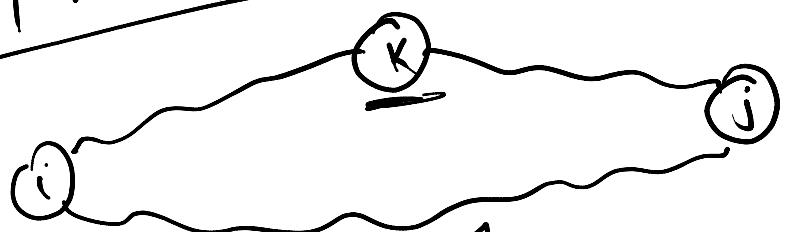
$$\begin{aligned} & \text{dis}(n+1)(n+1) \\ & d(i, j) \\ & d(i, i) = 0 \end{aligned}$$

for ( $k=0$  ;  $k < n$  ;  $k++$ )

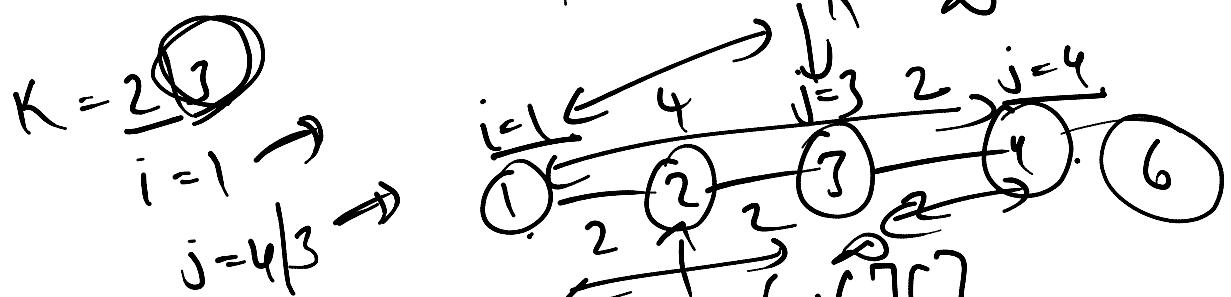
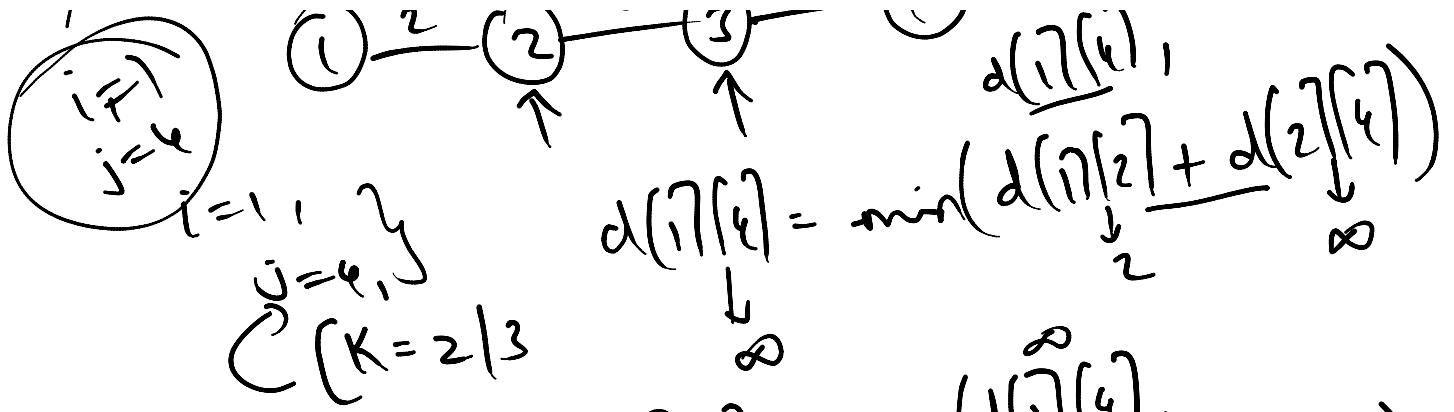
for ( $i=0$  ;  $i < n$  ;  $i++$ )

for ( $j=0$  ;  $j < n$  ;  $j++$ )

$$d(i, j) = \min(d(i, j), d(i, k) + d(k, j))$$

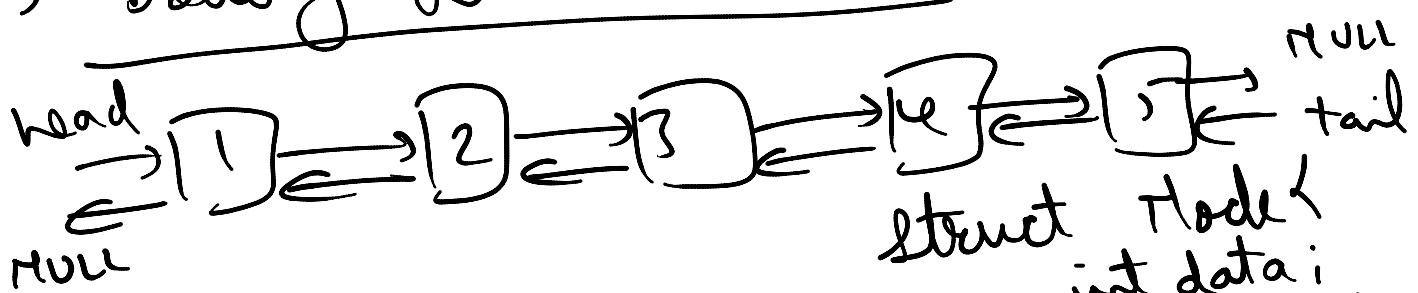


$$d(i, j) = \infty$$



$T C = O(n^3)$

$\Rightarrow$  Doubly Linked List :-



struct Node {  
 int data;  
 Node \*next;  
 Node \*prev;};

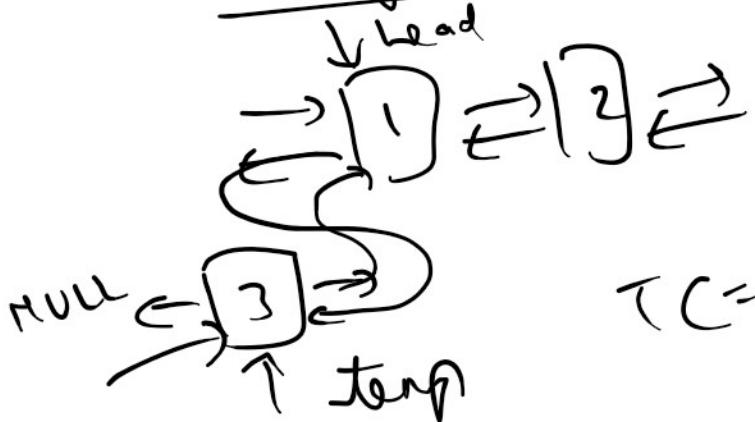
$\Rightarrow$  Insertion :-

Time :- 1

Ques :-

Head = tail

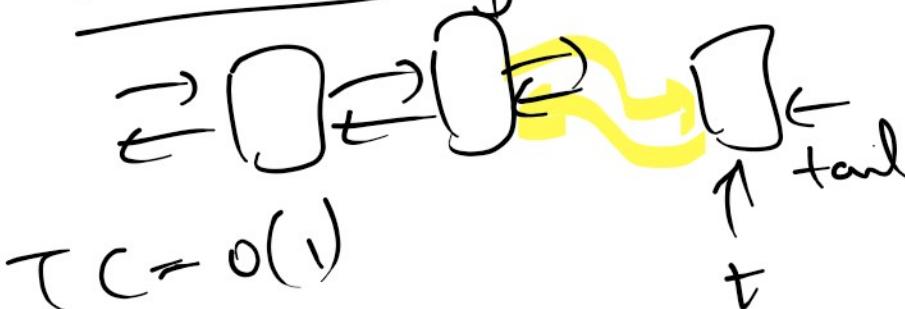
① at front position :-



$$TC = O(1)$$

~~temp  $\rightarrow$  next = head~~  
~~head  $\rightarrow$  next = temp~~  
~~temp head = temp~~

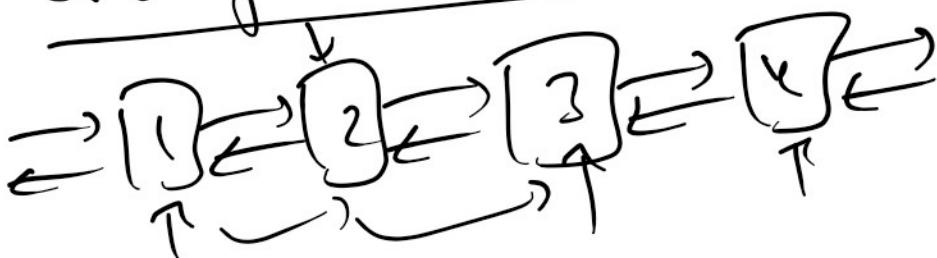
② at end :- tail



~~tail  $\rightarrow$  next = t~~  
~~t  $\rightarrow$  next = tail~~  
~~tail = t;~~

$$TC = O(1)$$

③ at given position :-



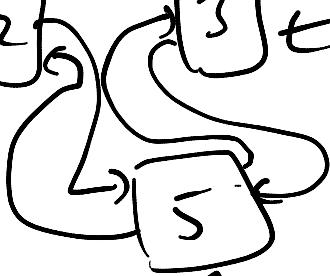
pos, head (pos-1)  
 pos-2  
 t

$$TC = O(N)$$



last = temp

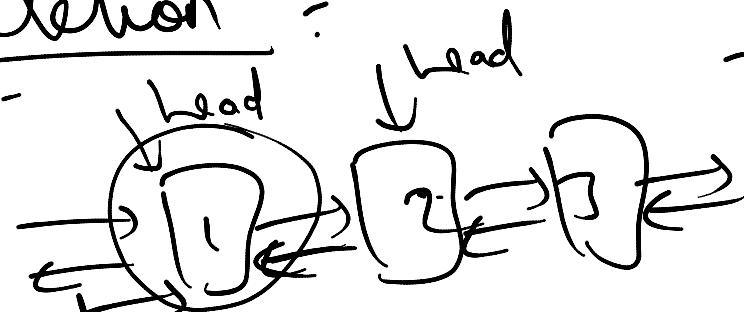
$$1 \rightarrow 2 \rightarrow 3 \leftarrow \text{NULL}$$



$t \rightarrow \text{next} \rightarrow \text{prev} = \text{temp}$   
 $\text{temp} \rightarrow \text{next} = t \rightarrow \text{next};$   
 $t \rightarrow \text{next} = \text{temp}$   
 $\text{temp} \rightarrow \text{prev} = t;$

$\Rightarrow$  Deletion :-

① Head :-



TC = O(1)

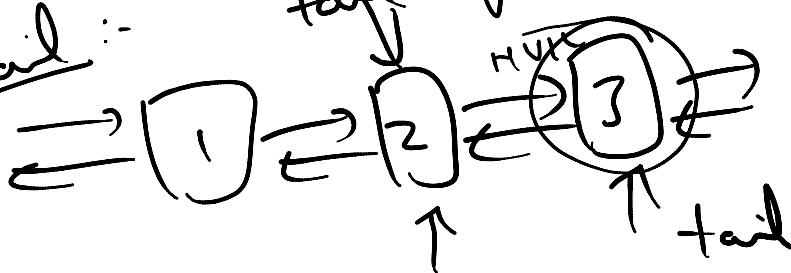
$\text{temp} = \text{head}$

$\text{head} = \text{head} \rightarrow \text{next};$

$\text{head} \rightarrow \text{prev} = \text{NULL};$

$\text{tail} \text{ free}(\text{temp});$

② Tail :-



TC = O(1)

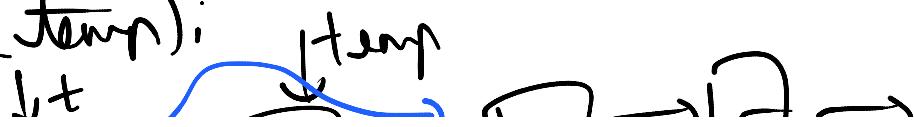
$\text{temp} = \text{tail}$

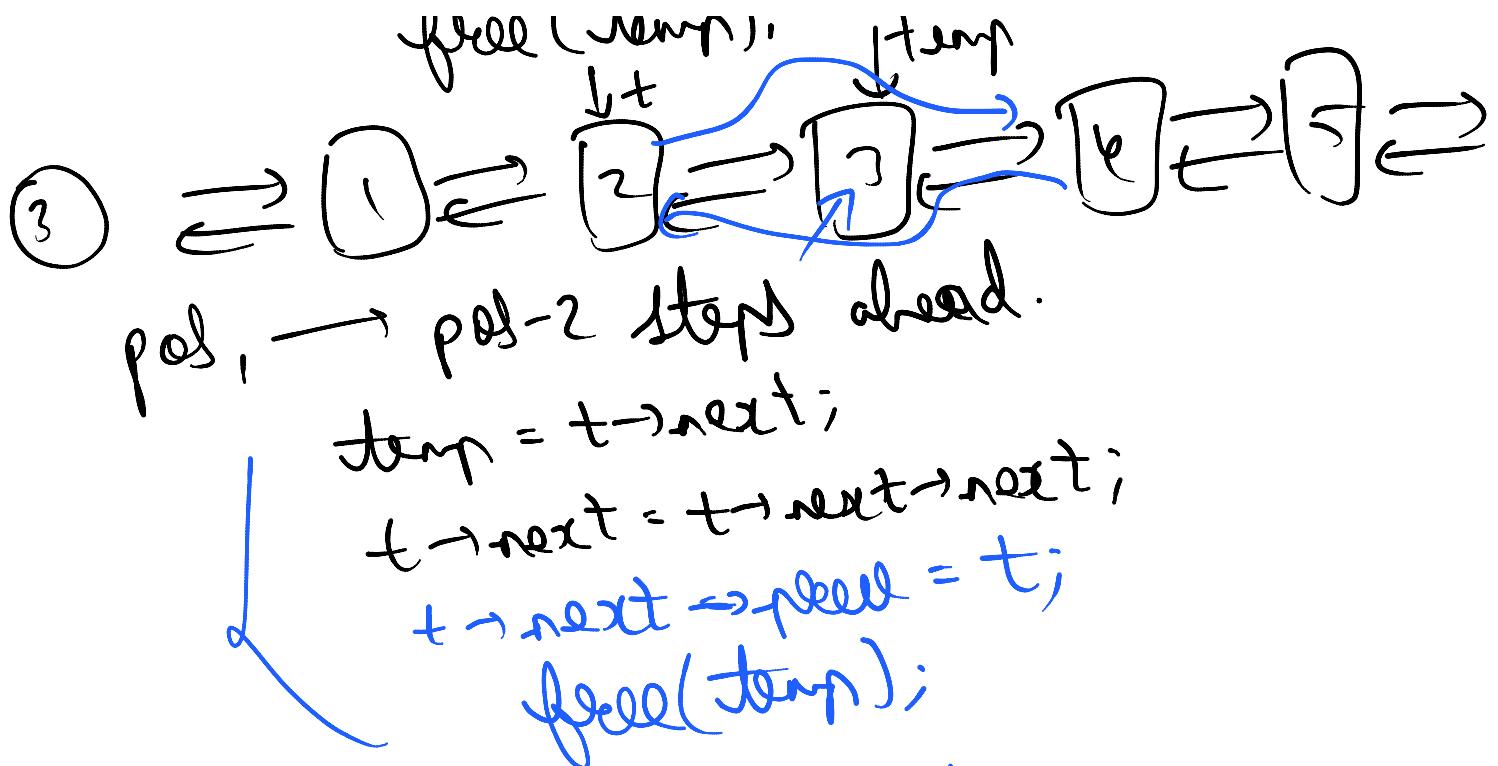
$\text{tail} = \text{tail} \rightarrow \text{prev};$

$\text{tail} \rightarrow \text{next} = \text{NULL};$

$\text{free}(\text{temp});$

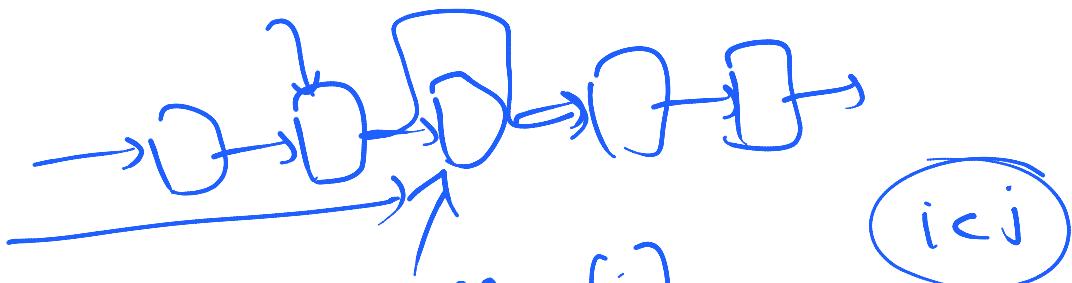
if tail is given.





$TC = O(n) \rightarrow \text{position}$

$= O(1) \rightarrow$  pointer is given to  
the node to be deleted.

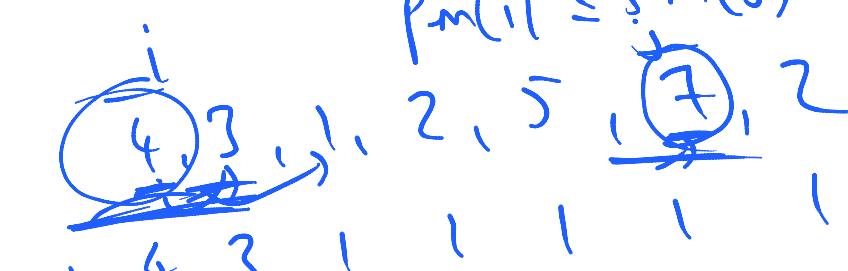


$\rightarrow i=0, j=0$

~~$p_{\min}(i) \leq p_{\max}(j)$~~

$p_{\min}(i) \leq p_{\max}(j)$

$p_{\min}(i)$



$j \rightarrow \dots$

$\rightarrow 4 \ 3 \ 1 \ 2 \ 5$   
 $\rightarrow 7 \ 7 \ 7 \ 2 \ \frac{2}{\infty} \ p_{\max}(i)$

$\rightarrow 7 \ 7 \ 7 \ 2 \ \frac{2}{\infty} \ p_{\max}(i)$

```

→  $\sum_{i=0}^n \sum_{j=0}^i$  if (pmin(i)  $\leq$  smax(j))
    { ord = max(ord, j-i);
      j++;
    }
    i++;
  }
}

```

$$\begin{array}{r}
 0, 1, 2, 3, 4, 5 \\
 \boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5} \quad \boxed{6} \\
 \hline
 1 & 2 & 3 & 4 & 5 & 6 \\
 \hline
 1 & 2 & 3 & 7 & 8 & 9 \\
 \hline
 1 & 2 & 3 & 4 & 3 & 1 \\
 \hline
 1 & 2 & 9 & 8 & 7 & 2 \\
 \hline
 2 & 1 & 3 & 4 & 2 & 1 \\
 \hline
 9 & 8 & 7 & 6 & 5 & 4
 \end{array}$$

→ Tree

struct node {  
 Node \* child[10];  
 int endpoint; };

