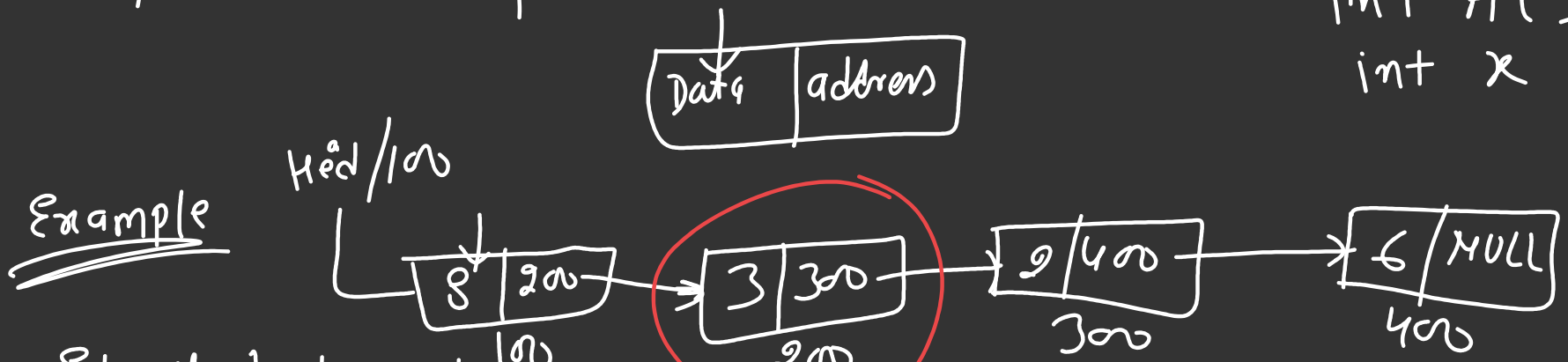
Date
28/12/22

§Data Structures§

Singly linked list :: This list consists of nodes in which each node has a data/value and next pointer to the following element.

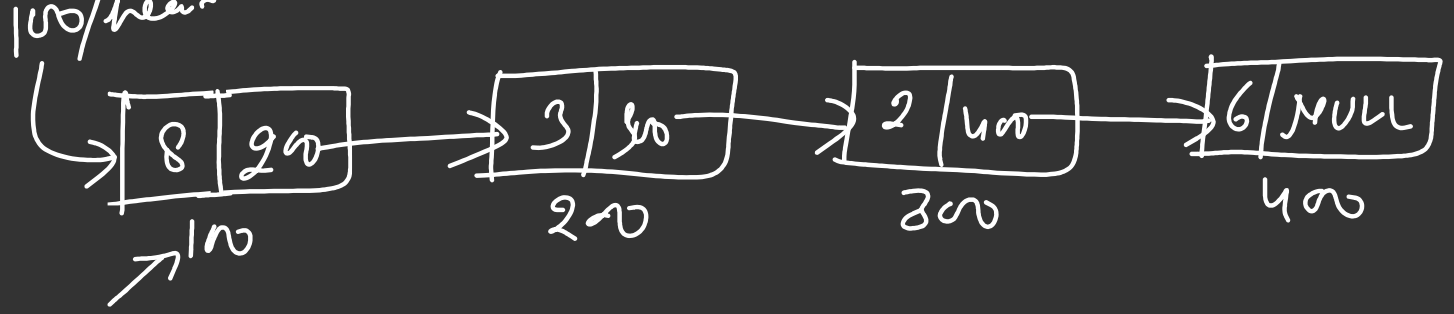
int A[]
int x




```
Struct list_node
{
    int data;
    Struct listNode *next;
};
```

Basic operations on linked list:

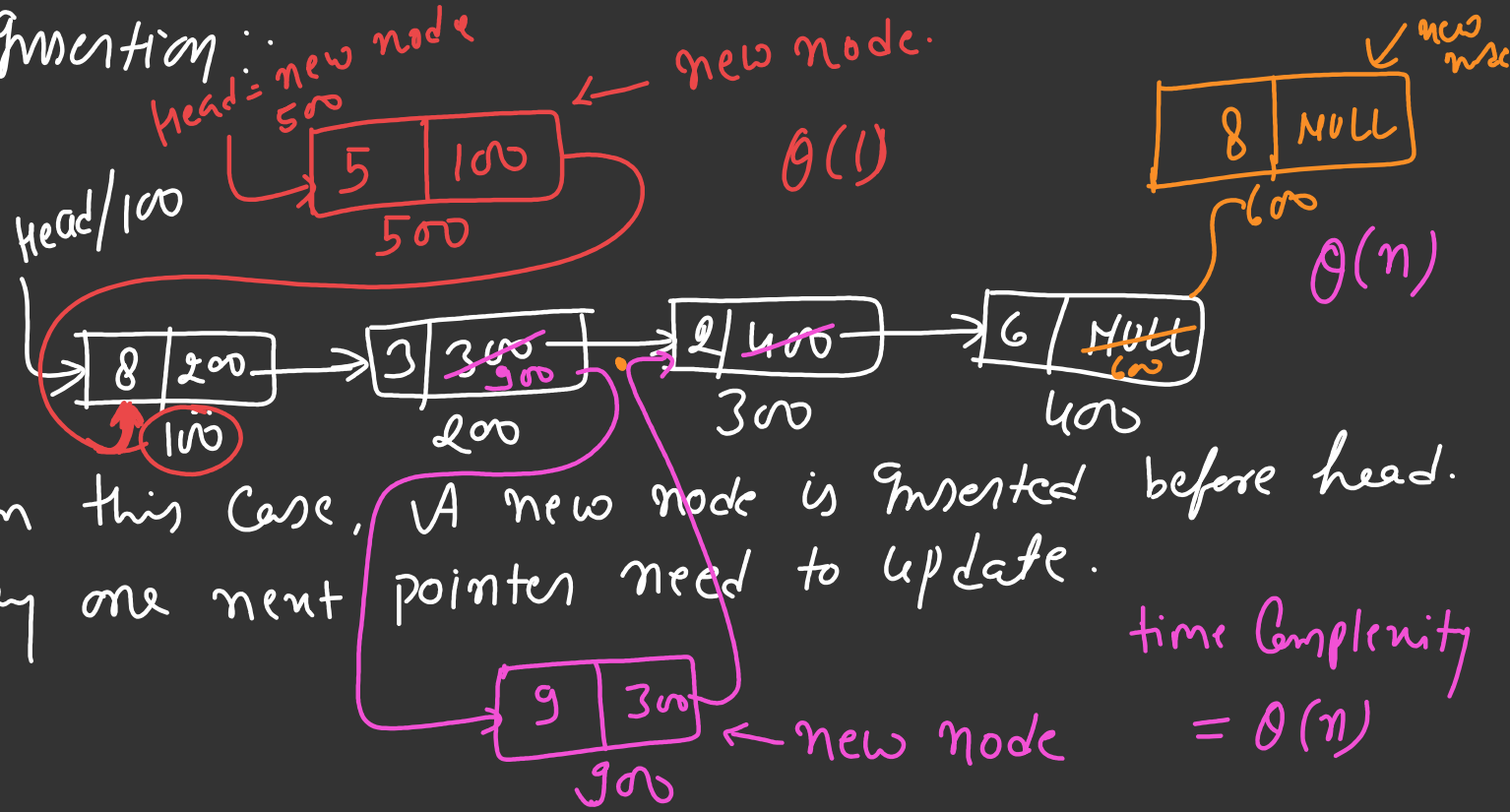
- Insertion
- Deletion
- Traverse




```
Traverse ( Struct list_node *head)
{
    Struct list_node *current
    Current = head;
    While (Current != NULL)
    {
        Print ("Current→data)
        Current = Current→next;
    }
}
```

Singly linked list insertion:

- At beginning
- At end.
- At intermediate.



↳ At beginning: In this case, A new node is inserted before head. Only one next pointer need to update.

time Complexity = O(n)

InsertInLinkedList (*head, int data, int position)

{

newnode=(struct node*)malloc(sizeof(struct node));

Struct node *p, *q, *newnode; ↳ Create a node.

newnode → data = data

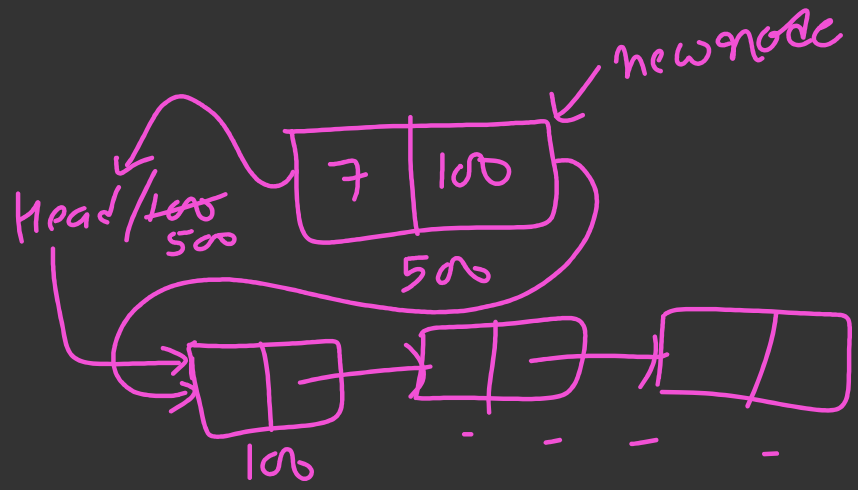
p = head;

/* Inserting at beginning */

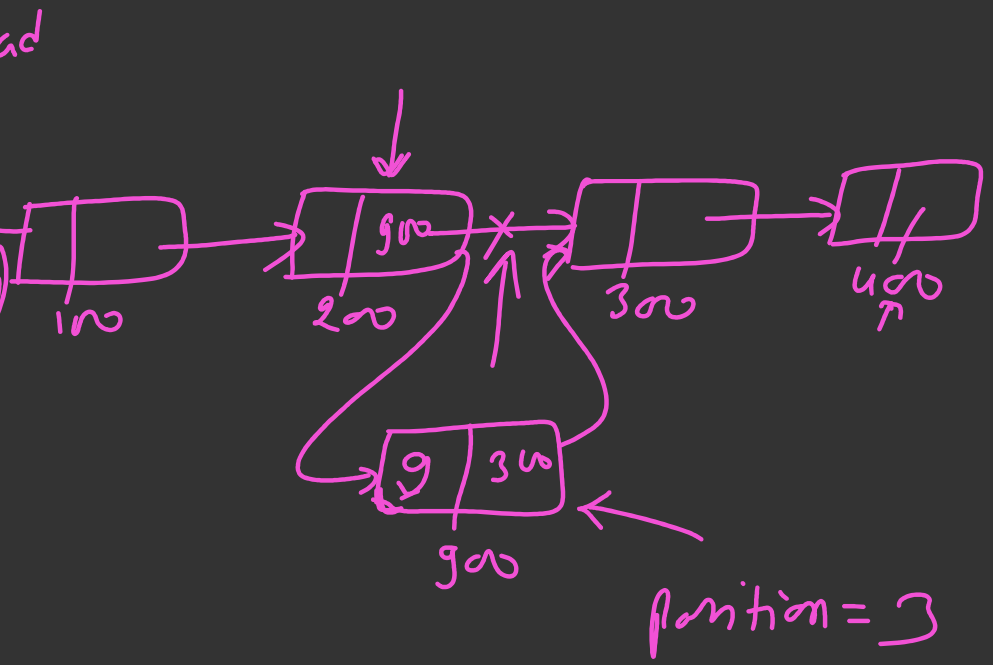
if (position == 1)

newnode → next = p;

head = newnode;



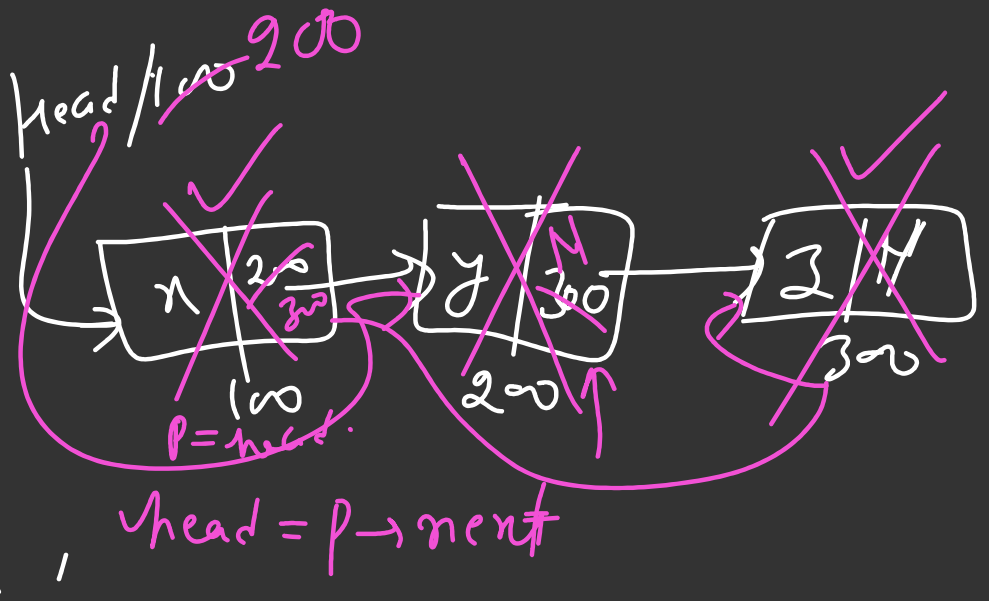
```
p=head;
K=1
else
{
while(p->next!=NULL && K<position)
{
    K++;
    q=p;
    p=p->next;
}
q->next=newnode;
newnode->next=p;
}
```



§ Singly linked list Deletion:

. beginning
. End
. Intermidiate

↳



Major = 70 (7 questions) ← sub parts — 3 hours

mid/minor = 30/15 (1 hour)

Assignment + viva = 30/25

} 100

Submit ↑ — 10 marks.

min = 5