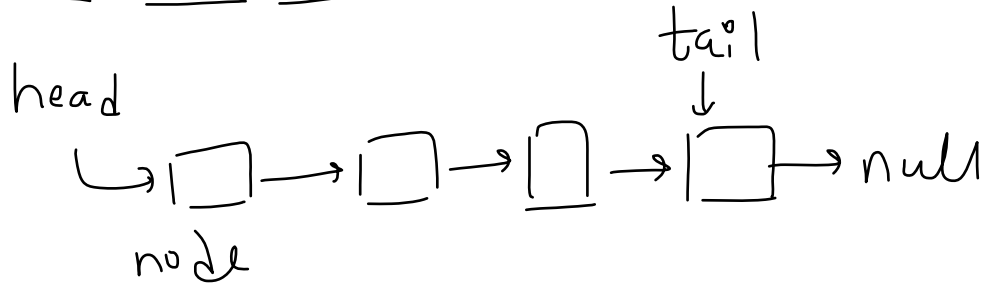
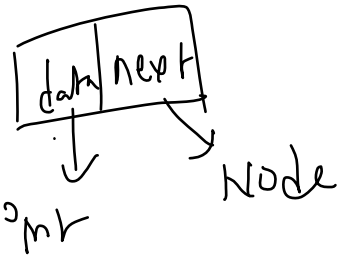
Single Linked List



node

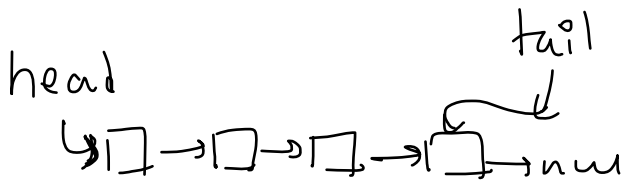


Operations

- insert
  - insertAtBeginning
  - insertAtEnd
  - insertInBetween
  - insertAtnthNode
- delete
- search
- sort
- traverse
- merge

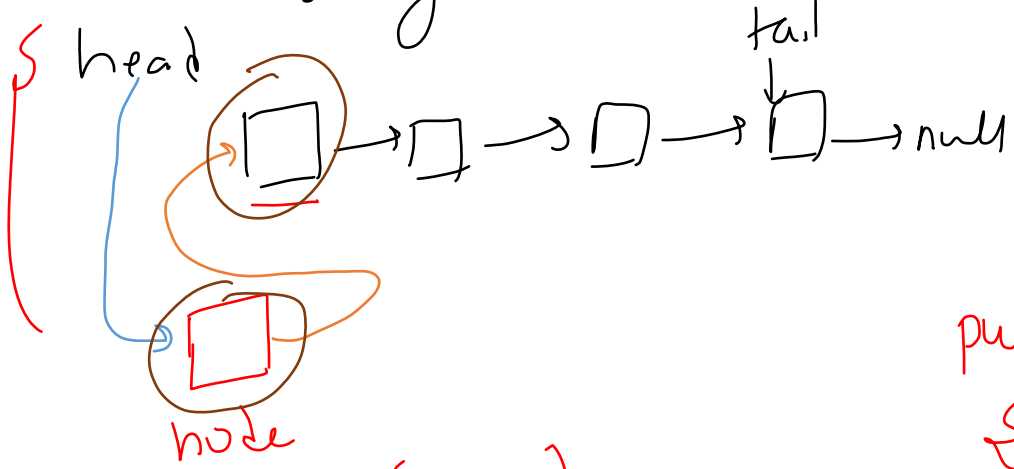
Size → total no of nodes in Linked List

Insert




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

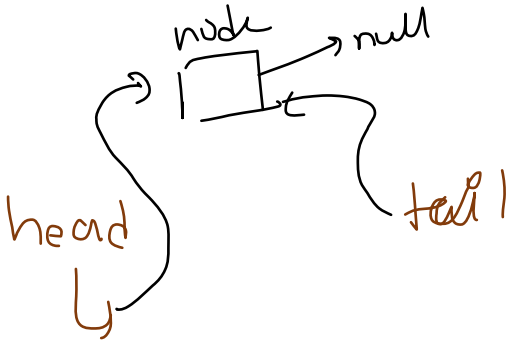
InsertAtBeginning / insertAtHead / addFirst




```
public Node
    (int data)
{
    this.data = data;
    next = null;
}
```

```
node.setNext(head);
head = node;
```

```
public void setData(int
    data)
{
    this.data = data;
}
public void setNext(Node
    next)
{
    this.next = next;
}
```

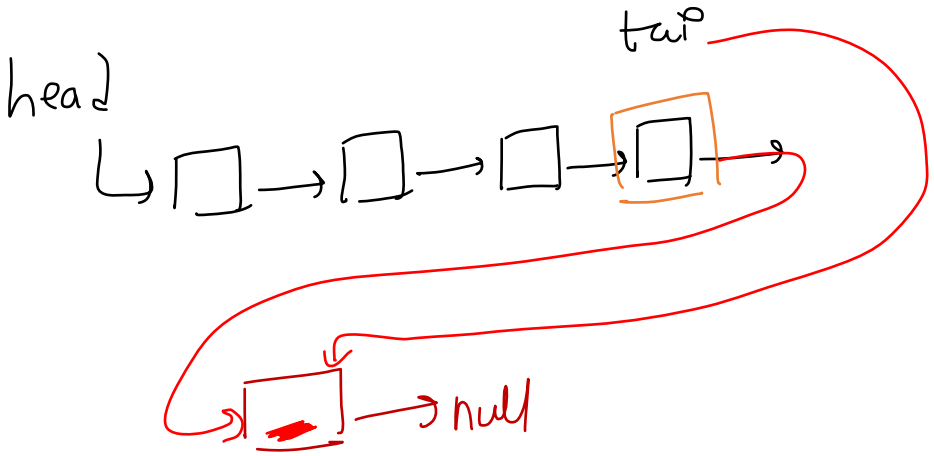



```
if (isEmpty())
{
    head = node;
    tail = node;
}
```

```
if (head == null)
{
}
```

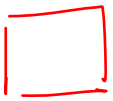
```
boolean isEmpty()
{
    if(head == null)
    {
        return true;
    }
    return false;
}
```

insertAtEnd / insertAtLast / insertAtTail / addLast



node

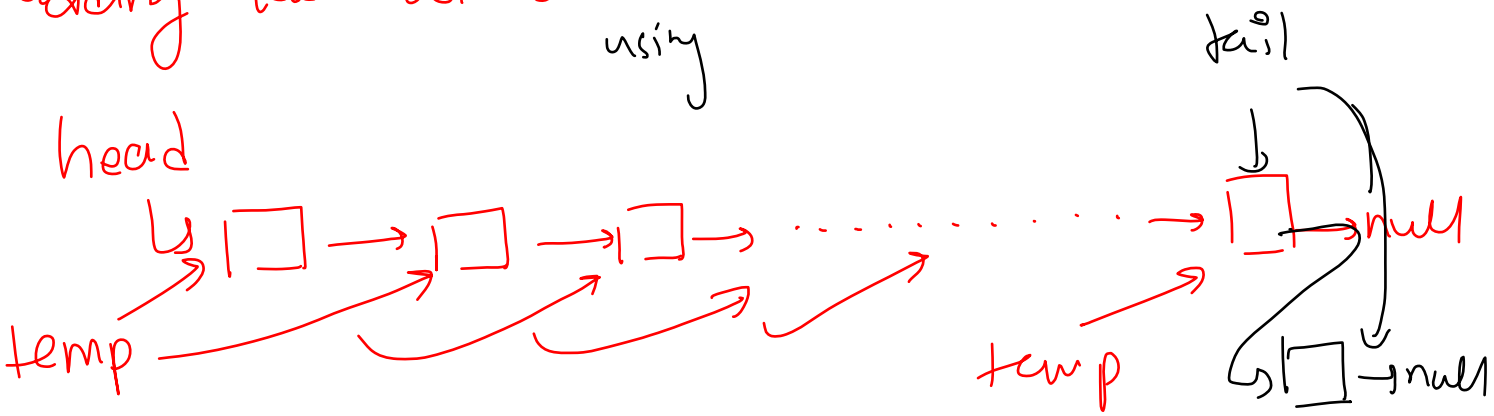
```
tail.setNext(node);
tail = node;
```




```
if (isEmpty())
{
    head = node;
    tail = node;
}
```

addFirst()

adding last without tail using



temp = temp.getNext();

else

node

```
Node temp = head;
while ( temp.getNext() != null)
{
    temp = temp.getNext();
}
temp.setNext (node);
}
```