

# DSA through Java

## Doubly linked list

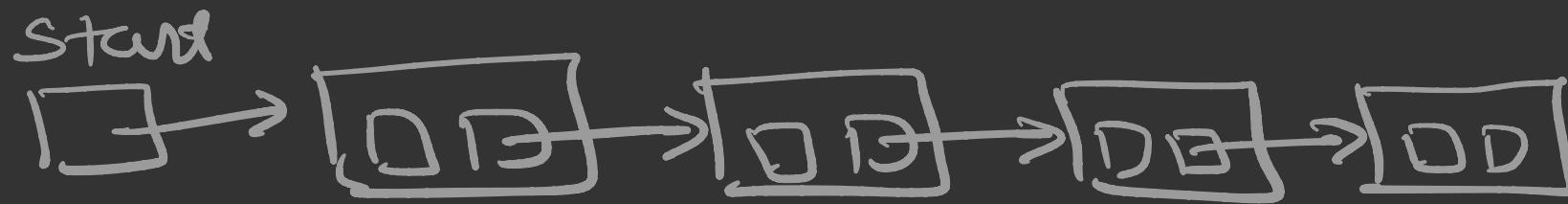


Saurabh Shukla (MySirG)

## Agenda

- ① Shortcomings of singly linked list
- ② Doubly linked list
- ③ node
- ④ insertion
- ⑤ deletion

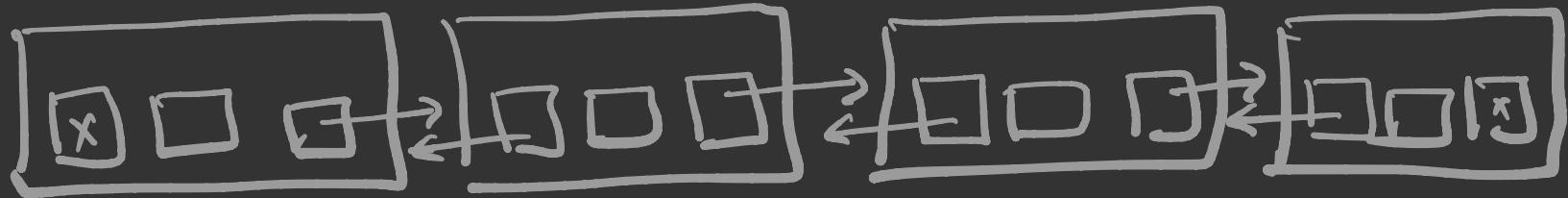
## Shortcomings of Singly linked list



In SLL, you can move only in the forward direction

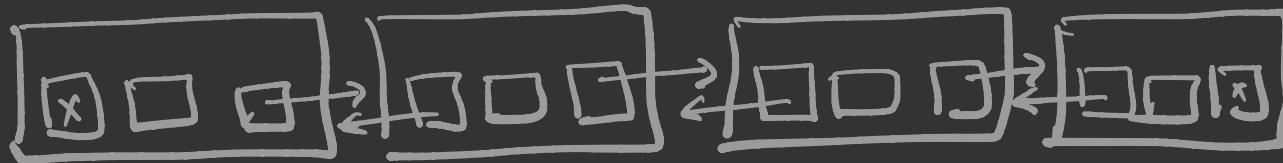
# Doubly linked list

Start



# node

Start



class node

{

    node prev;

    int item;

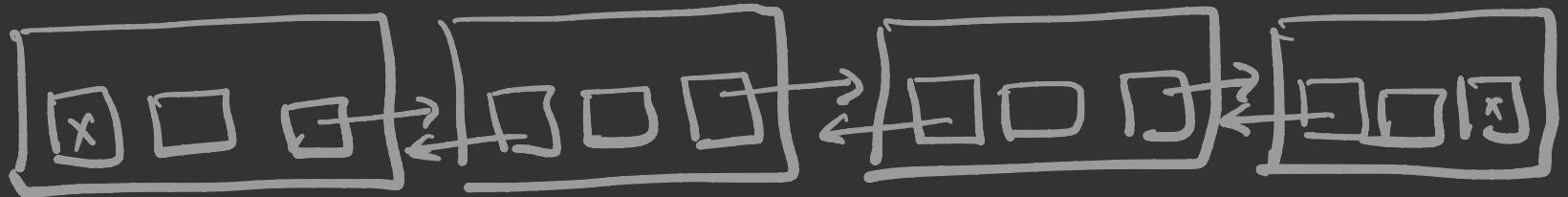
    node next;

}



# Insertion

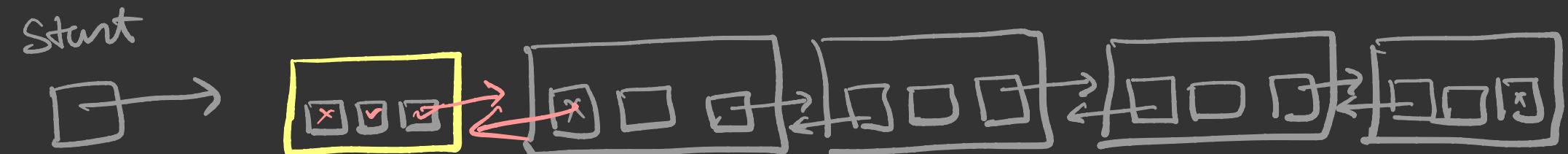
Start



## Insertion

- ① At First
- ② At Last
- ③ After a node

## Insert as a first node



```
node n = new node();
```

```
n.item = data;
```

```
n.prev = null;
```

```
n.next = start;
```

```
if (start != null)
```

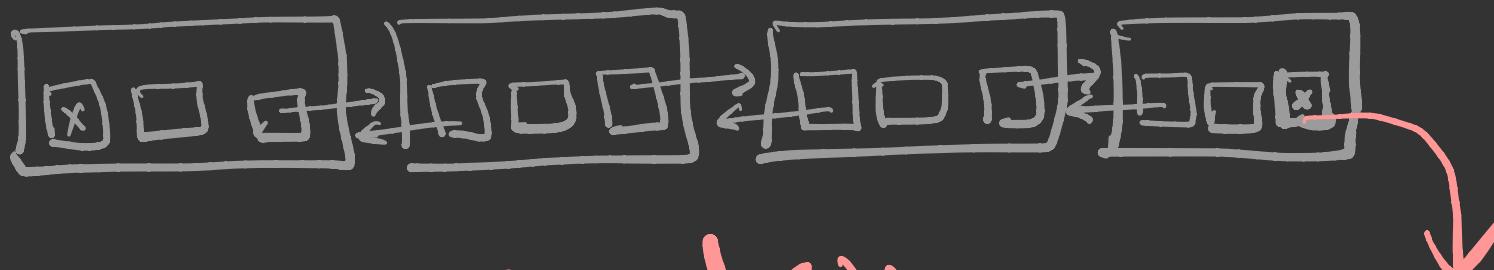
```
    start.prev = n;
```

```
start = n;
```



## Insert at last

Start



temp  
temp

```
node n = new node();
```

```
n.item = data;
```

```
n.next = null;
```

```
if (start == null)
```

```
{
```

```
    n.prev = null
```

```
    start = n;
```

else  
{}

```
temp = start;
```

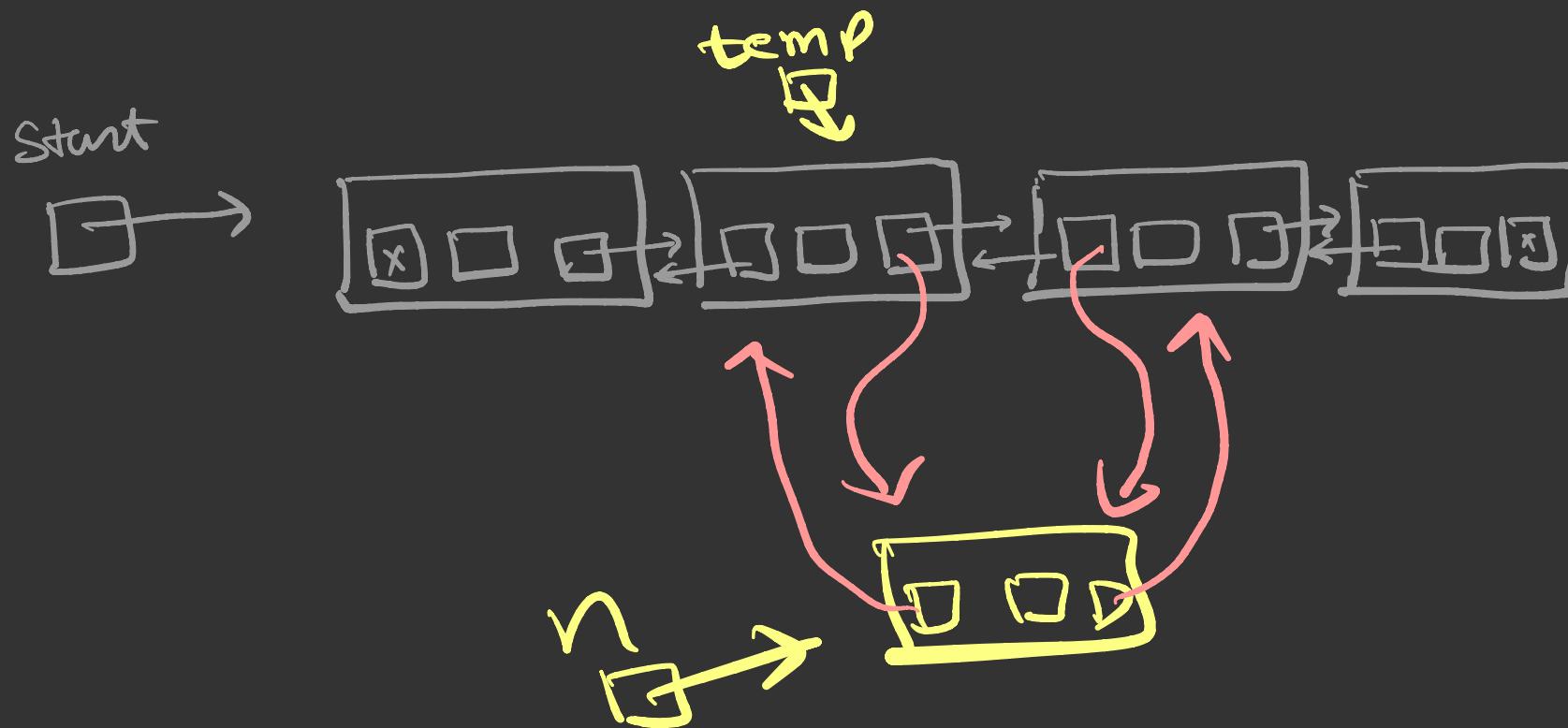
```
while (temp.next != null)
```

```
    temp = temp.next;
```

```
    temp.next = n;
```



# Insert After a node



## Deletion

- ① Delete first node
- ② Delete Last Node
- ③ Delete specific node