

# C++ LAUNCHPAD



CODING  
BLOCKS

Lecture-15,16

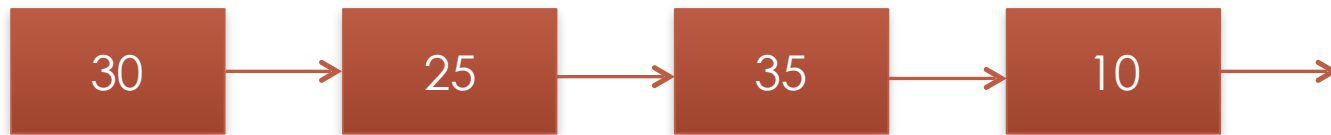
## Data Structures

- Linked List- I
- Linked List - II

Utkarsh Nath

# What are Data Structures?

# What are Linked Lists?



# Lets define our own Linked List

```
struct Node {  
    int data;  
    Node* next;  
};
```

# Head and Tail nodes

# Basic operations over Linked List

1. Taking Linked List as input from user
2. Taking Linked list input, creating new head
3. Accessing next element from a given node
4. Looping over Linked List
5. Inserting at position k in Linked List
6. Deleting at position K in Linked List

# Lets try some problems

- Find length of a linked list
- Find mid point of a linked list
- Find an element recursively
- Implement Bubble Sort

## Lets try some problems

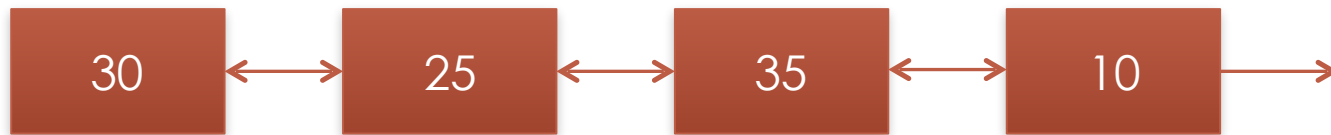
- Find  $i^{\text{th}}$  element from end without calculating length of Linked List
- Given two sorted linked lists merge them into a sorted linked list
- Implement merge sort
- Reverse a Linked List



# More problems

- Cycle Detection
- K- Reverse

# Doubly Linked Lists



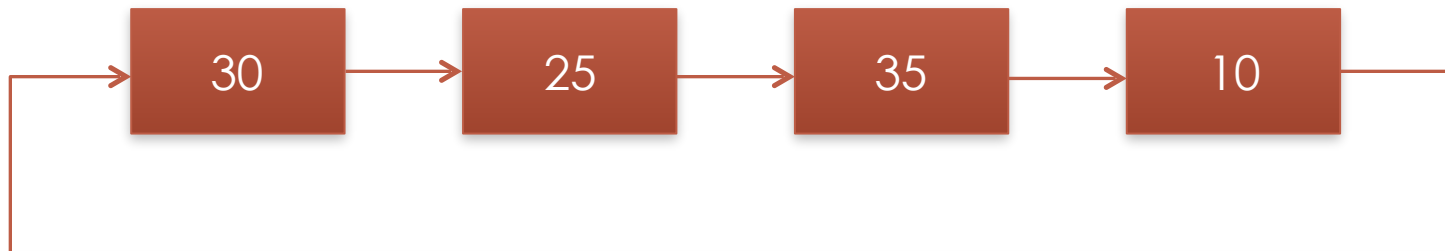
# Implementation?

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

# Doubly LL vs Singly LL

1. Faster to go back in the linked list
2. Uses more memory

# Circular Linked Lists



# Benefits of Arrays over Linked List

1. Random access to elements
2. Fast iteration through the elements
3. Very compact way to store data

# Benefits of Linked List over Array

1. Constant time insertion and deletion of elements
2. Don't need to know the number of elements
3. Insert elements in the middle of the list

# C++ LAUNCHPAD



CODING  
BLOCKS

Thank You!

Utkarsh Nath