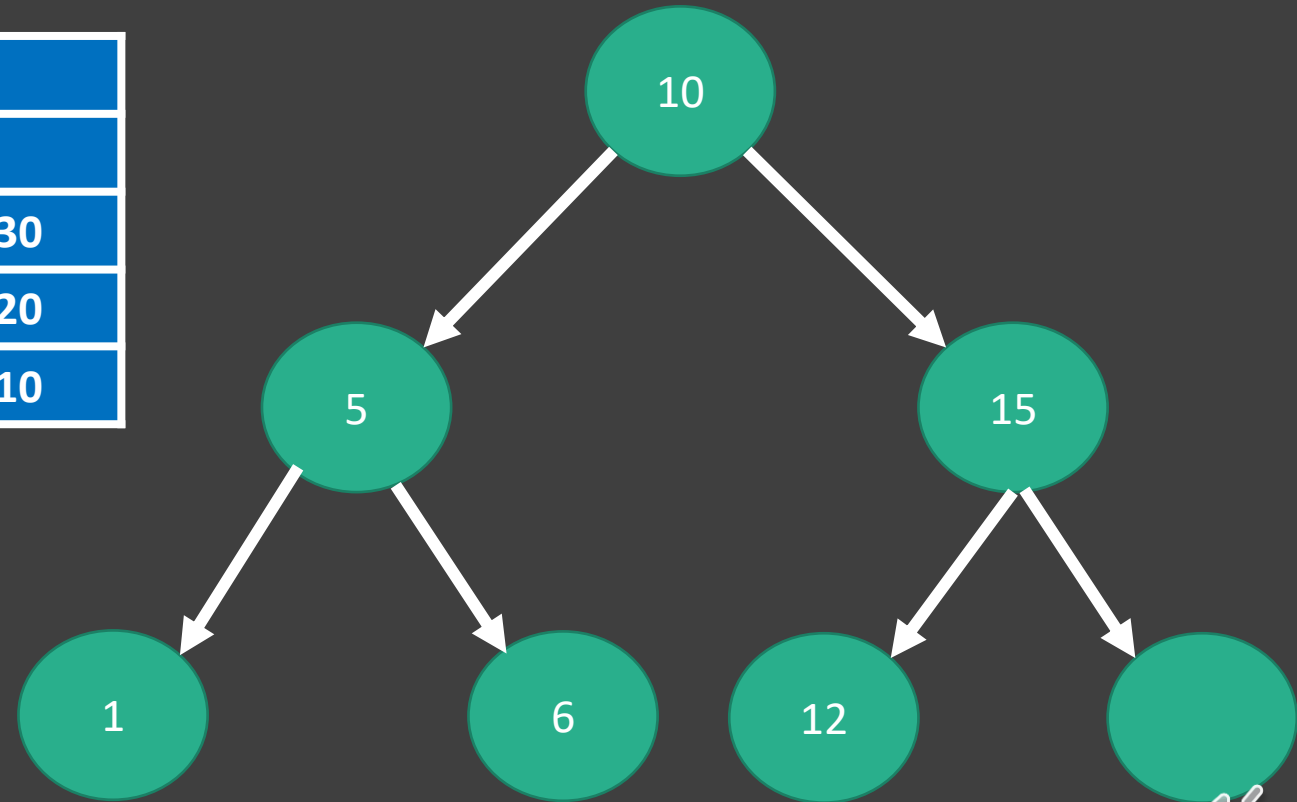
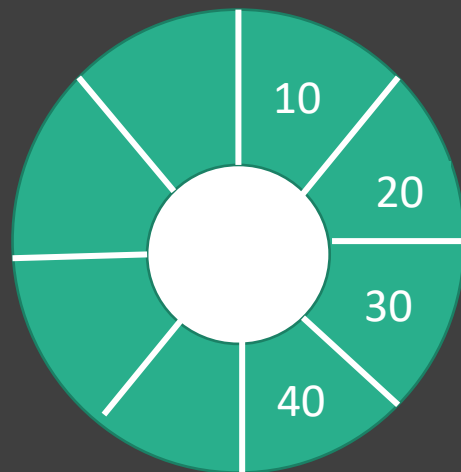
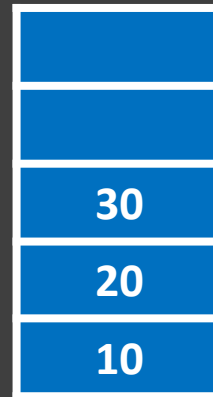
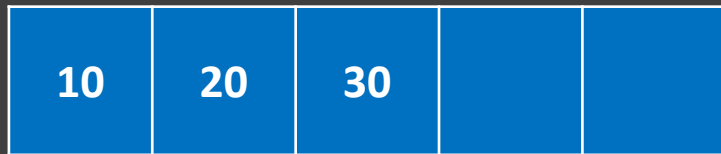
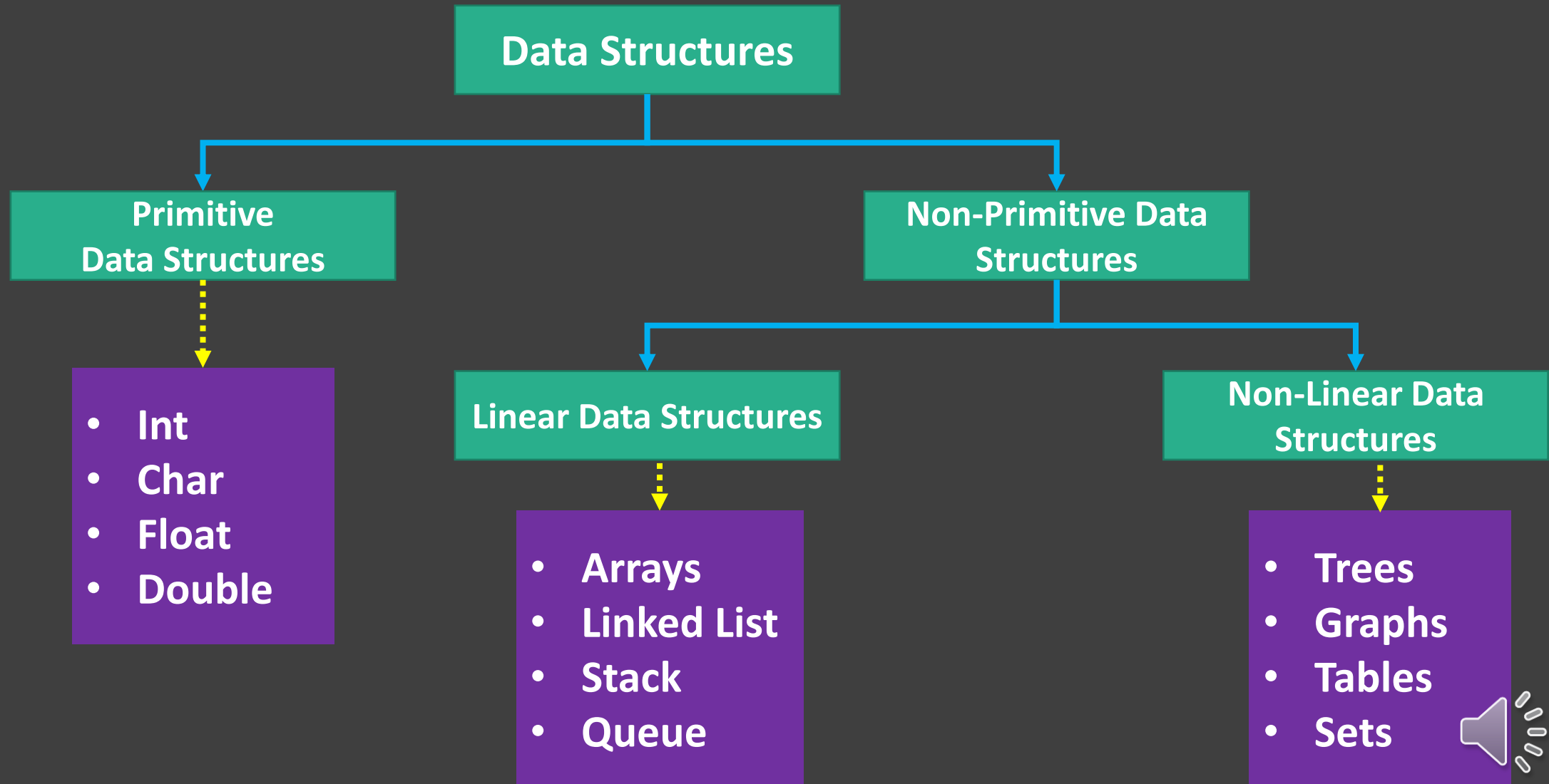


# Data Structures

Arranging individual data elements in an order to solve a particular problem. It can also be used to represent logical relationships between data elements.



# Types of Data Structures



# Primitive Data Structures

- ✘ These are Built in Data structures
- ✘ It can store single value
- ✘ Can not represent relationship between data elements

Exmple:

```
int roll1=20;
```

```
int roll2=30;
```

```
int roll3=40;
```

**Roll1=20**

Address: 65524

**Roll2=30**

Address: 65510

**Roll3=40**

Address: 76512

# Non-Primitive Data Structures

- ✓ It can store more than one value
- ✓ Can represent relationship between data elements

## Linear Data Structures

- ✓ Stores values in continuous order
- ✓ Easy to implement
- ✓ Access all the data values in single run

Ex: Arrays, Stack, Queue, Linked List

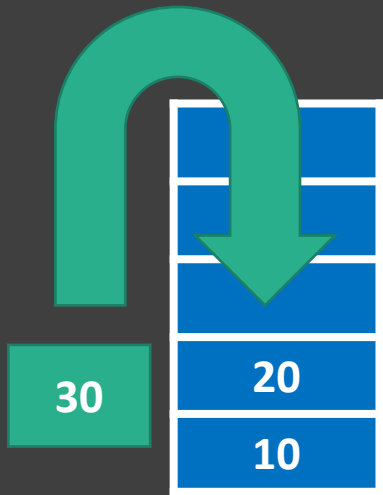
## Non-Linear Data Structures

- ✓ Stores values in non-continuous order
- ✓ Difficult to implement
- ✓ Can't access all the data values in single run
- ✓ Recursion is used to access data

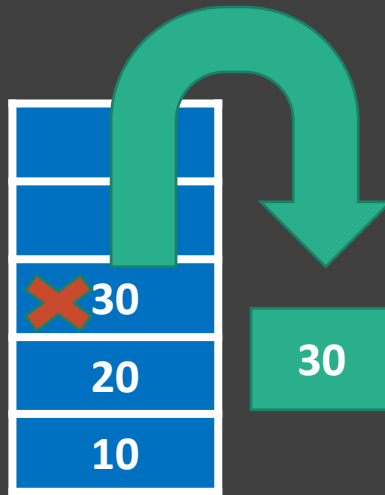
Ex: Trees, Graphs, Tables, Sets

# Operations on Data Structures

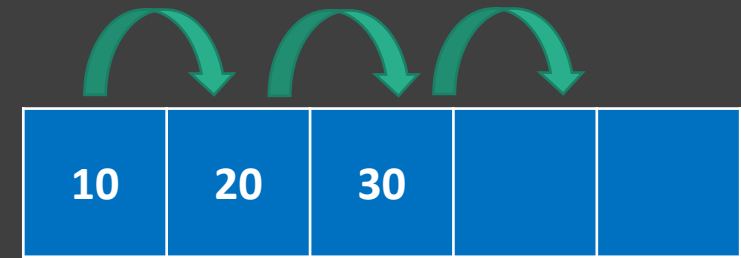
Insert



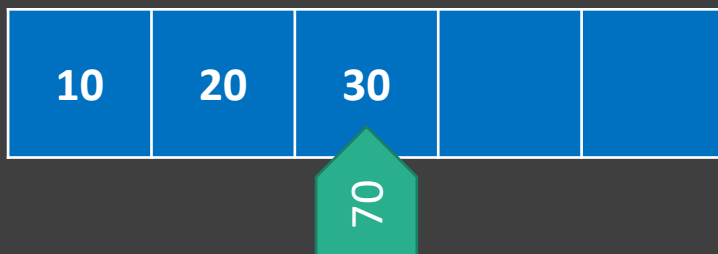
Delete



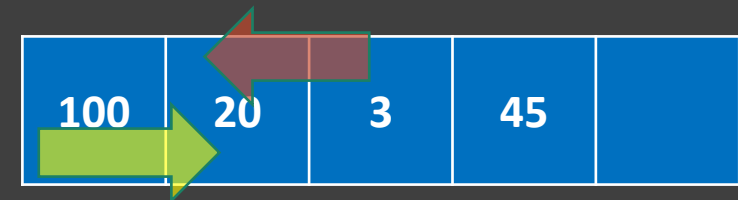
Searching



Updating



Sorting



Thank you