Sep22 : Day 1

Kiran Waghmare

CDAC Mumbai

Algorithms are Everywhere

- Search Engines
- GPS navigation
- Self-Driving Cars
- E-commerce
- Banking
- Medical diagnosis
- Robotics
- Algorithmic trading
- and so on …

Intelligent Computational Systems

"Big data" will allow us to put the "smarts" into everything …

- Smart homes
- Smart cars
- Smart health
- Smart robots
- Smart crowds and human-computer systems
- Smart interaction (virtual and augmented reality)
- Smart discovery (exploiting the data deluge)

Algorithm Design Strategies

- Brute force
- Divide and conquer
- Decrease and conquer
- Transform and conquer
- Greedy approach
- Dynamic programming
- Backtracking and branch and bound
- Space and time tradeoffs

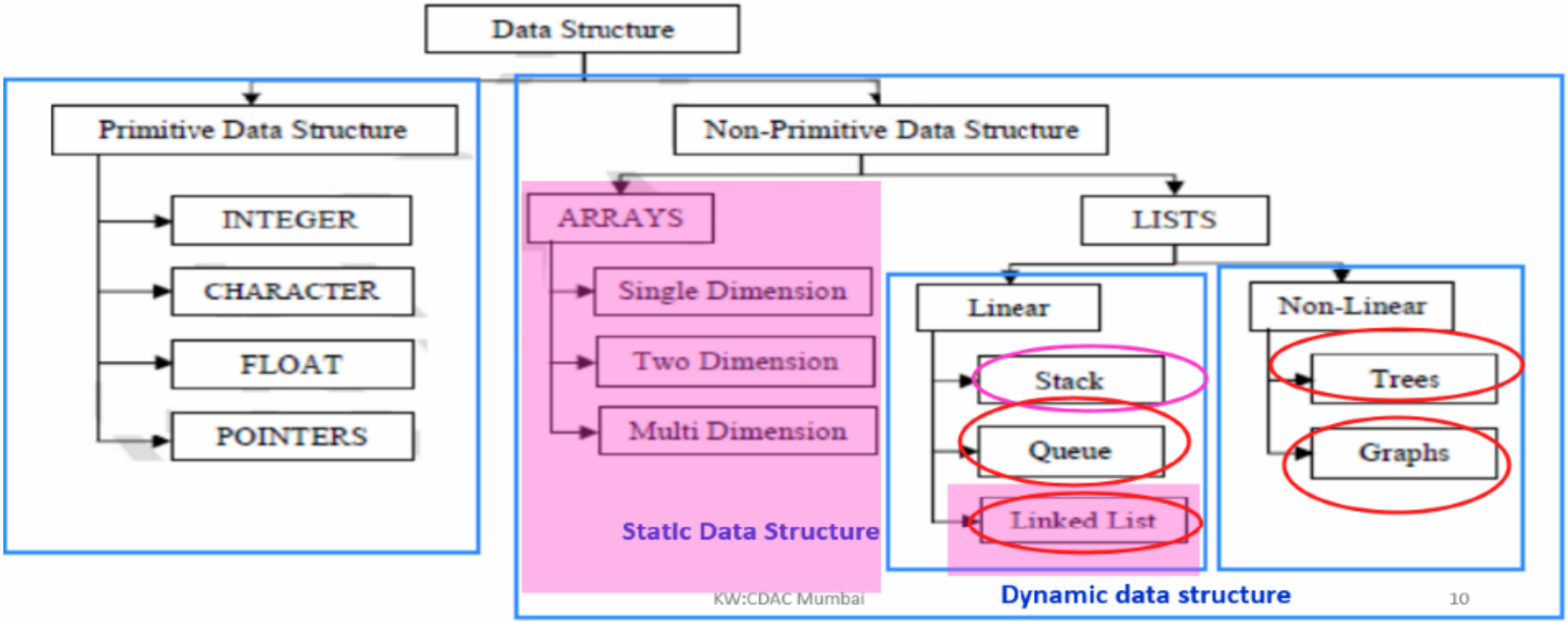
Invented or applied by many genius in CS

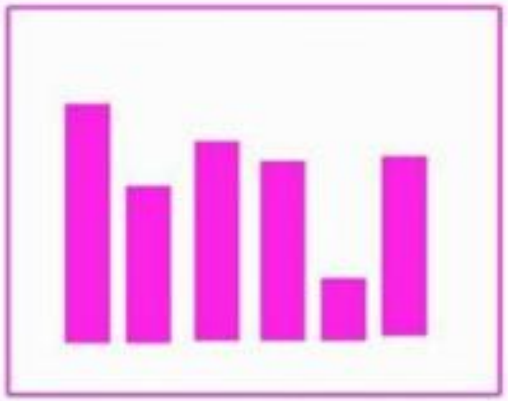
In general

- **A good algorithm is a result of repeated effort and rework**
  - Better data structure
  - Better algorithm design
  - Better time or space efficiency
  - Easy to implement
  - Optimal algorithm

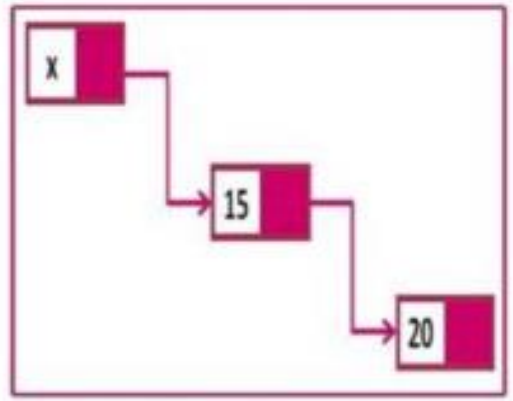
IMPLEMENTATION -> Arrays, Linked List

Classification of Data Structure

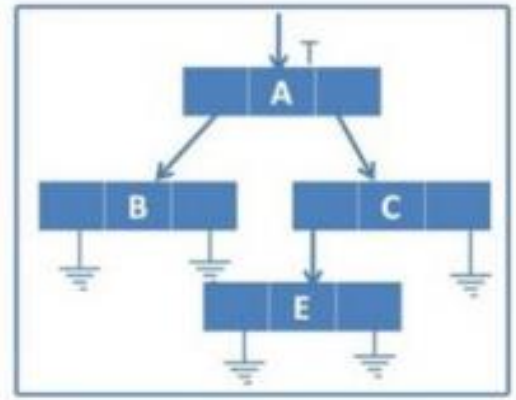




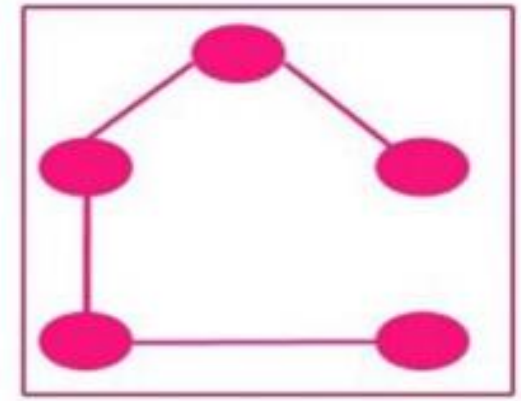
Sorting



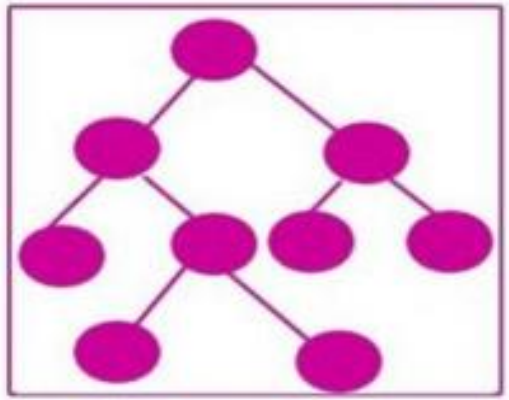
Link list



list

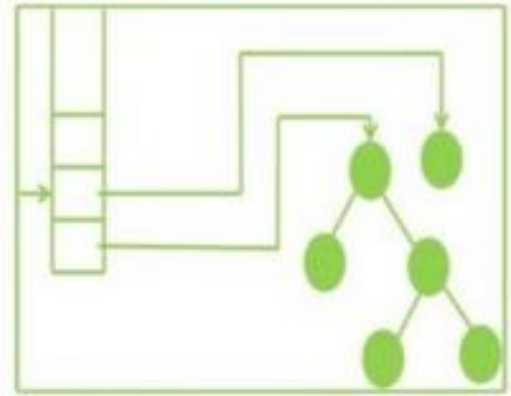


spanning tree

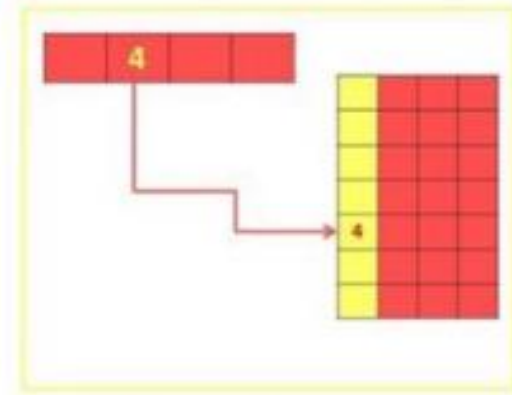


Tree

Graph



Stack



Hashing

By...navinkumardhoprephotography.com

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo

Who can see what you share here? Recording On

Date : 01/03/2023

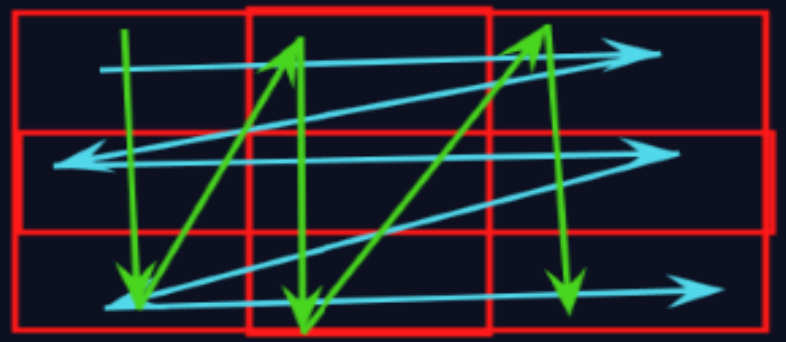
Day 1 : Python ADS

ADS: Algorithms + Data structure

linear: sequence :Array

Non-linear: not ina sequence : tree, graph

homogeneous:similar :



Row-major

R1
R12
R13

R21
R22
R23

R31
R32
R33

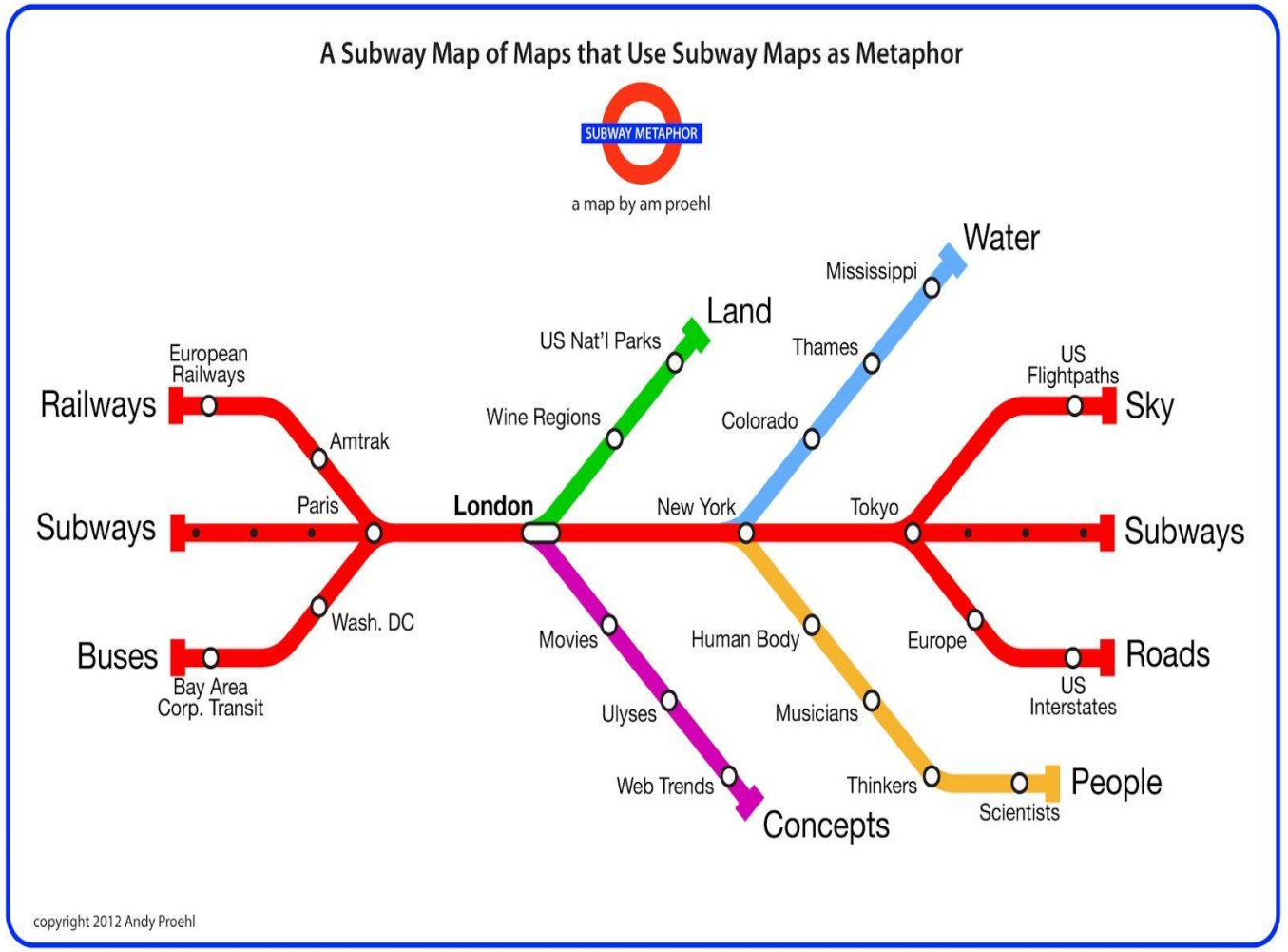
Col-major

c11
c12
c13

c21
c22
c23

c31
c32
c33


INDIA
Air Network
AFGHANISTAN
PAKISTAN
CHINA
NEPAL
BHUTAN
BANGLADESH
MYANMAR
Kabul
Srinagar
Leh
Amritsar
Dehradun
New Delhi
Kathmandu
Jaisalmer
Jaipur
Lucknow
Agra
Varanasi
Patna
Kolkata
Dibrugarh
Guwahati
Imphal
Ahmedabad
Bhopal
Jabalpur
Ranchi
Dhaka
Nagpur
Raipur
Bhubaneswar
Mumbai
Pune
Aurangabad
Hyderabad
Vijayawada
Visakhapatnam
Rajahmundry
Panaji
Bengaluru
Chennai
Puducherry
Port Blair
Kochi
Tiruchirappalli
Thiruvananthapuram
Colombo
SRI LANKA
Arabian Sea
Bay of Bengal
INDIAN OCEAN
LEGEND
International Route
Domestic Route
International Airport
Domestic Airport
Map not to Scale
398 × 470


A Subway Map of Maps that Use Subway Maps as Metaphor
SUBWAY METAPHOR
a map by am proehl
Railways
European Railways
Amtrak
Subways
Paris
Buses
Wash. DC
Bay Area Corp. Transit
London
Land
US Nat'l Parks
Wine Regions
Water
Mississippi
Thames
Colorado
New York
Concepts
Movies
Ulysses
Web Trends
Sky
US Flightpaths
Tokyo
Subways
Roads
US Interstates
Europe
People
Human Body
Musicians
Thinkers
Scientists
copyright 2012 Andy Proehl

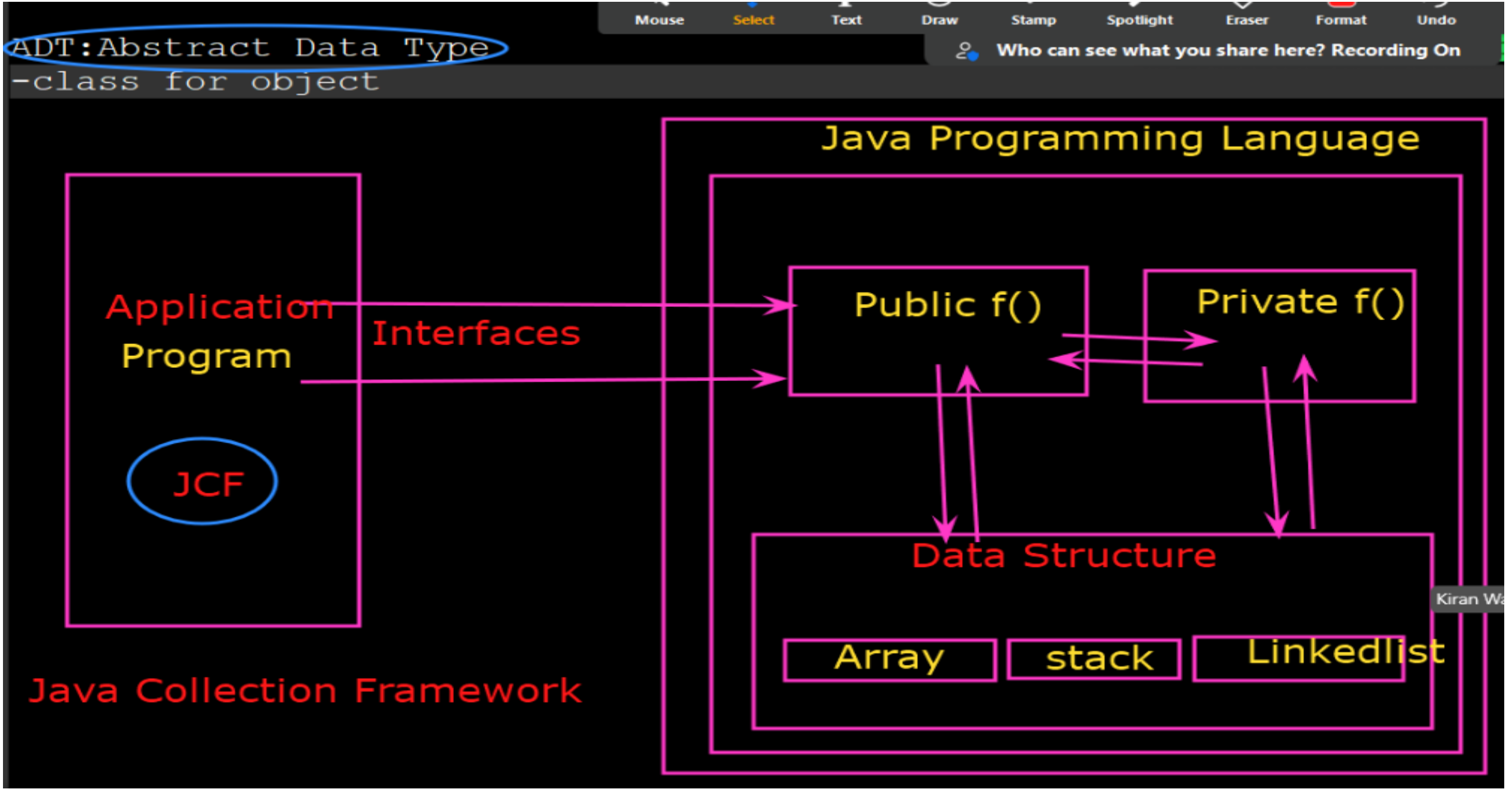
Abstract Data Type (ADT)

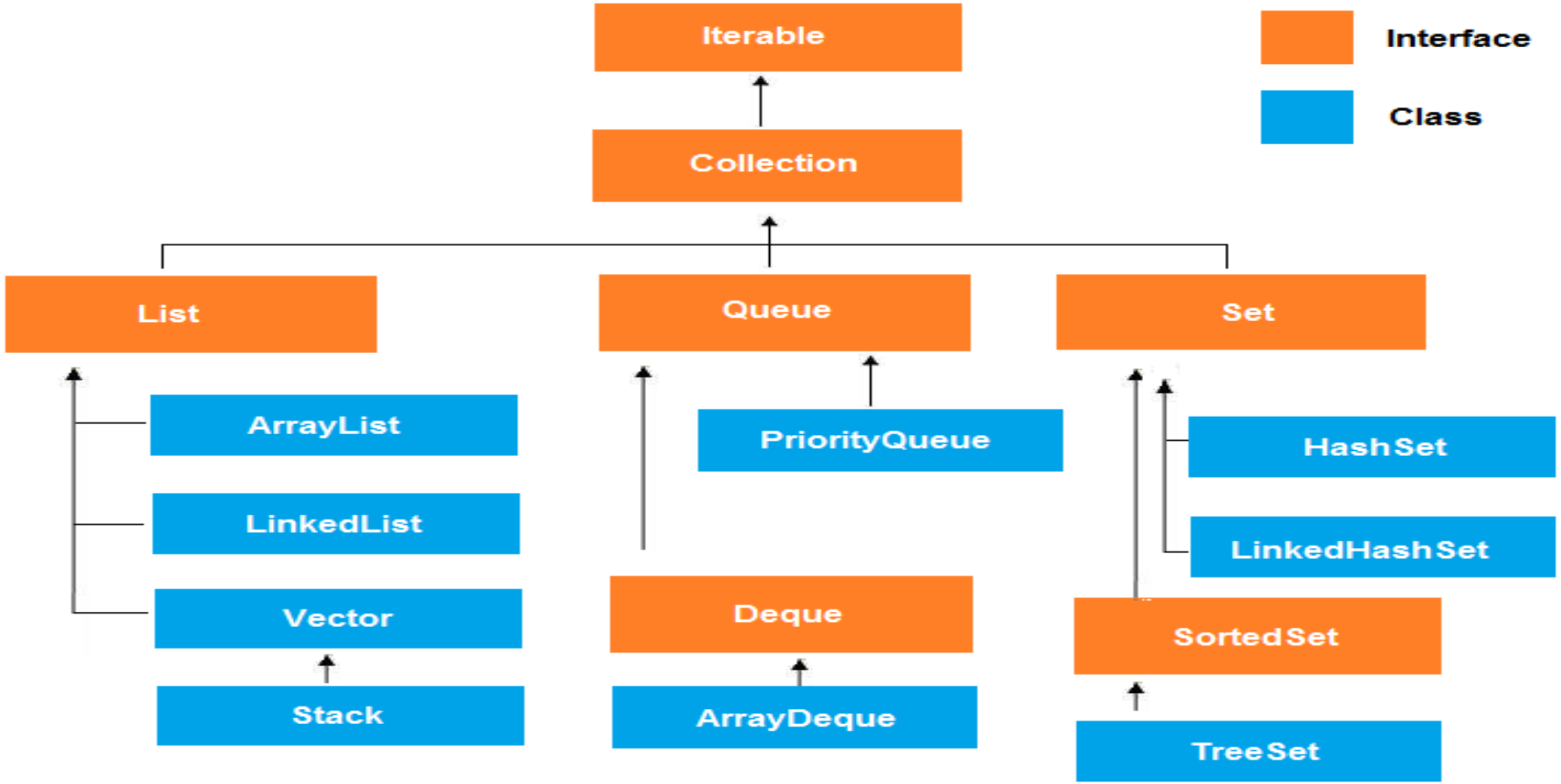
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo

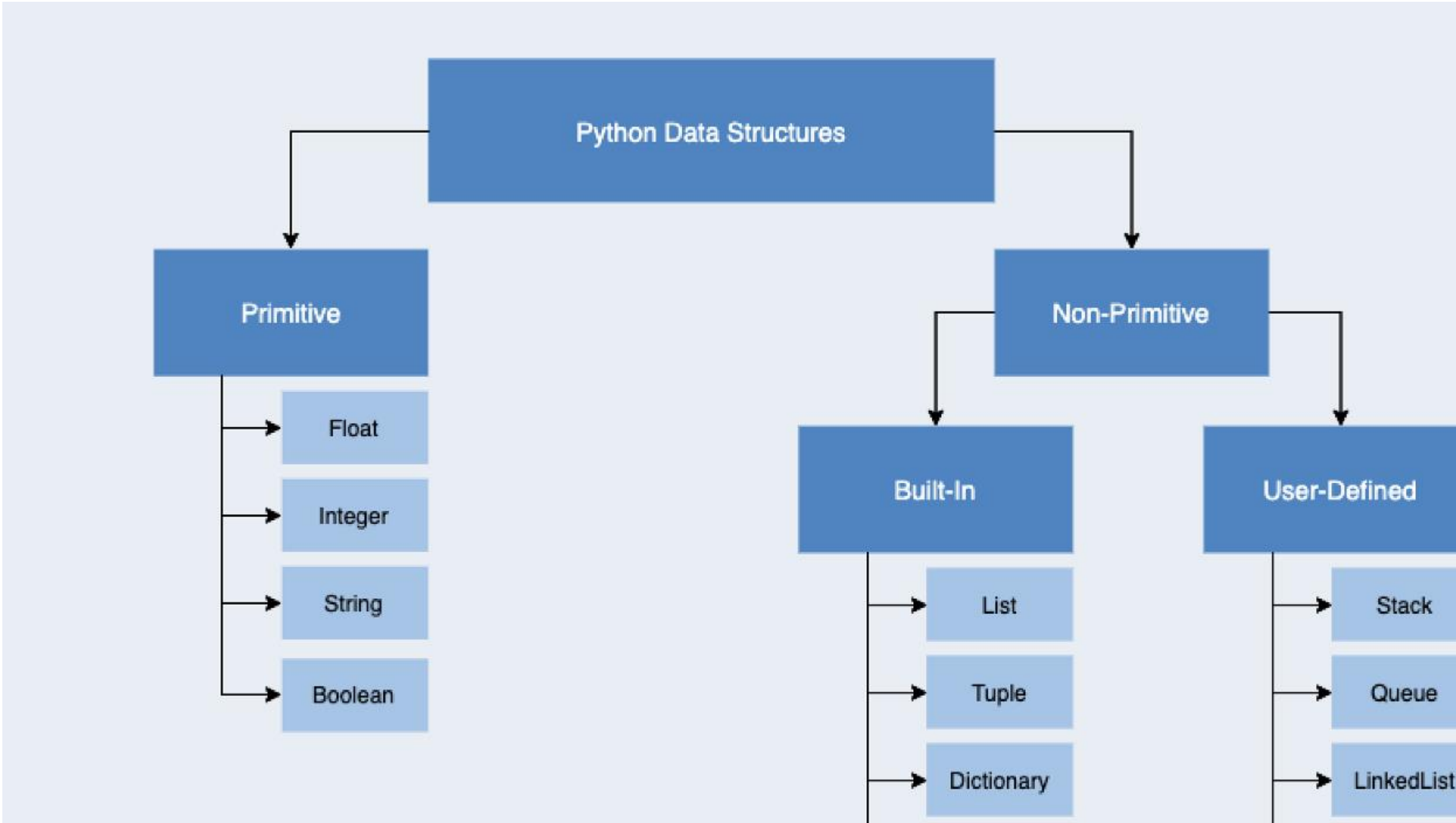
Who can see what you share here? Recording On

ADT:Abstract Data Type

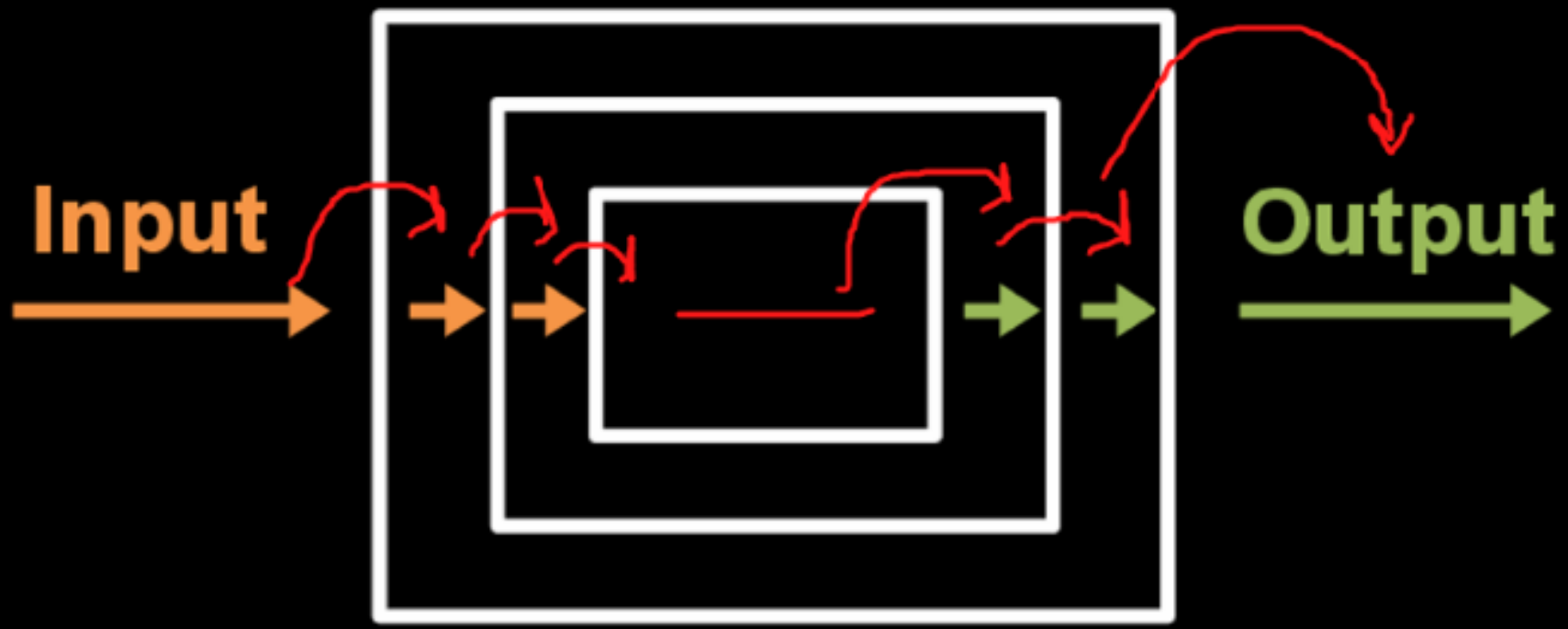
-class for object



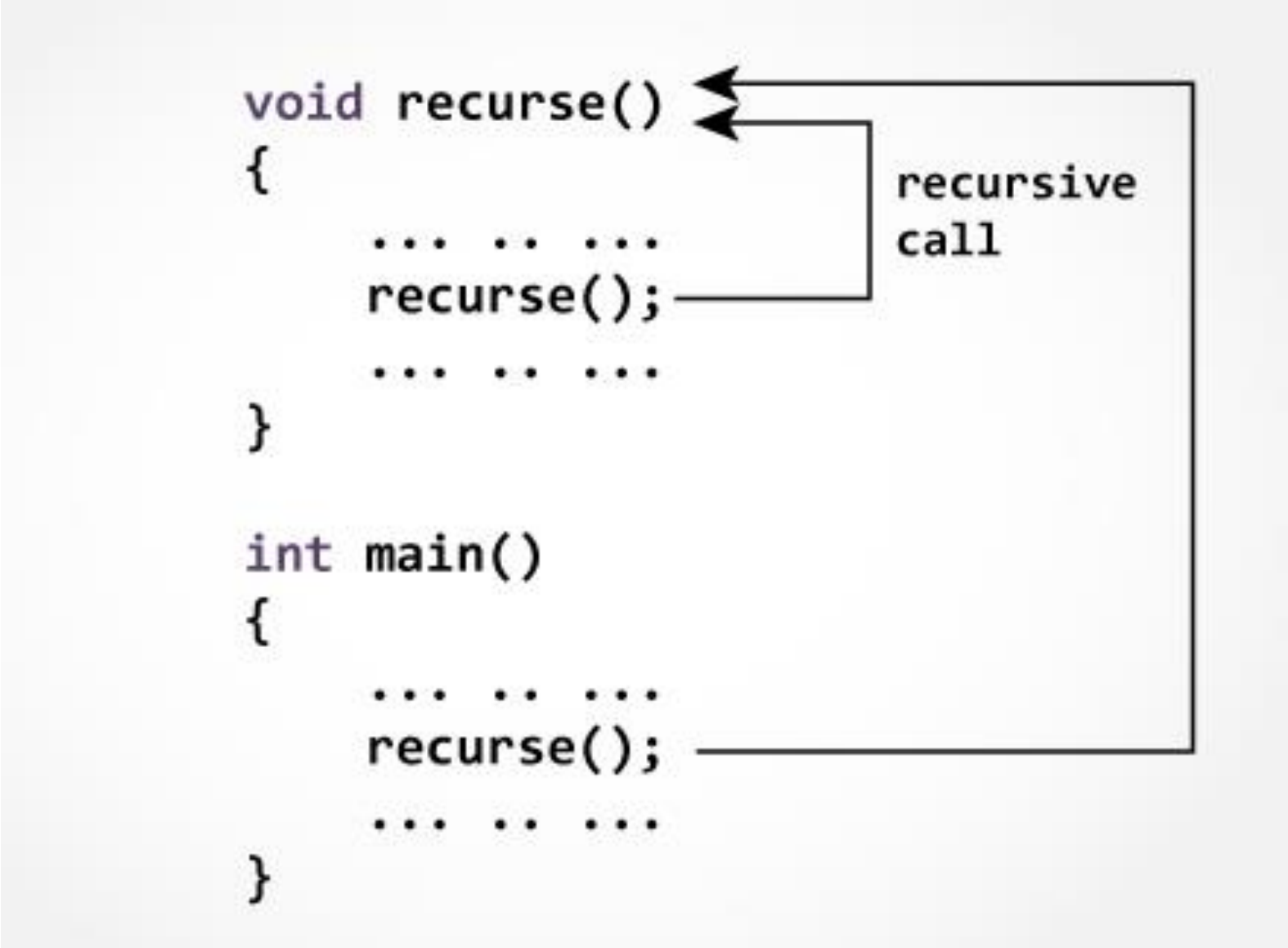

Iterable
Collection
List
Queue
Set
Interface
Class
ArrayList
LinkedList
Vector
Stack
PriorityQueue
Deque
ArrayDeque
HashSet
LinkedHashSet
SortedSet
TreeSet


Python Data Structures
Primitive
Float
Integer
String
Boolean
Non-Primitive
Built-In
List
Tuple
Dictionary
User-Defined
Stack
Queue
LinkedList

Recursion



How does Recursion works?



Direct recursion:

```
def Abc():

    // Some code....
    Abc();
    // Some code...
```

- **Indirect recursion:**

```
def Fun1():

    // Some code...
    Fun2();
    // Some code...

def Fun2()

    // Some code...
    Fun1();
    // Some code...
```

```
def abc():
    xyz()

def xyz():
    abc()
```


```
Example: 1+2+3+4+5
def recursive(num):
    if num == 0
        return 0
    else:
        return num + recursive(num-1)
```

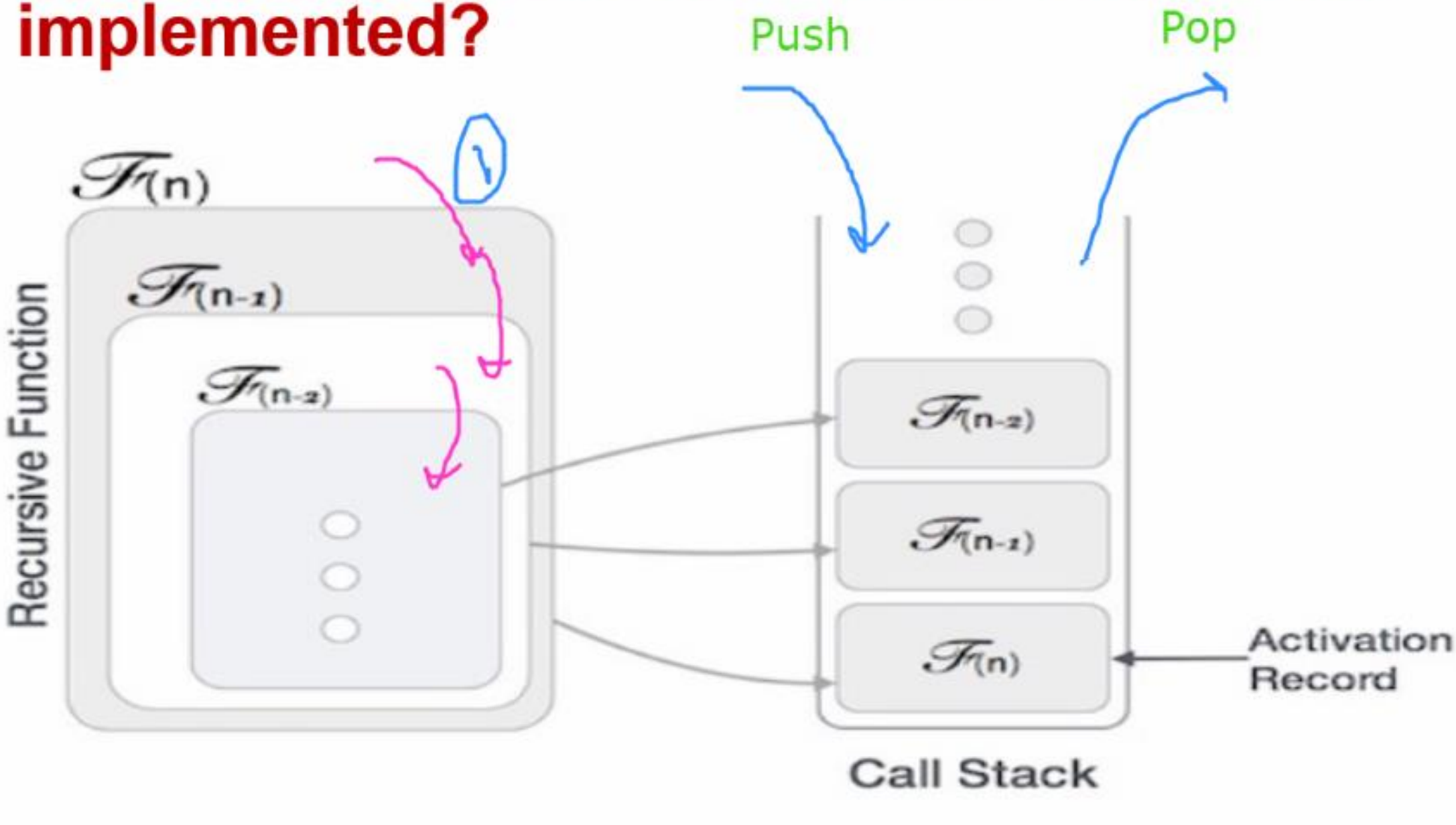
Problem 1: Write a program and recurrence relation to find the Fibonacci series of n where n>2 .

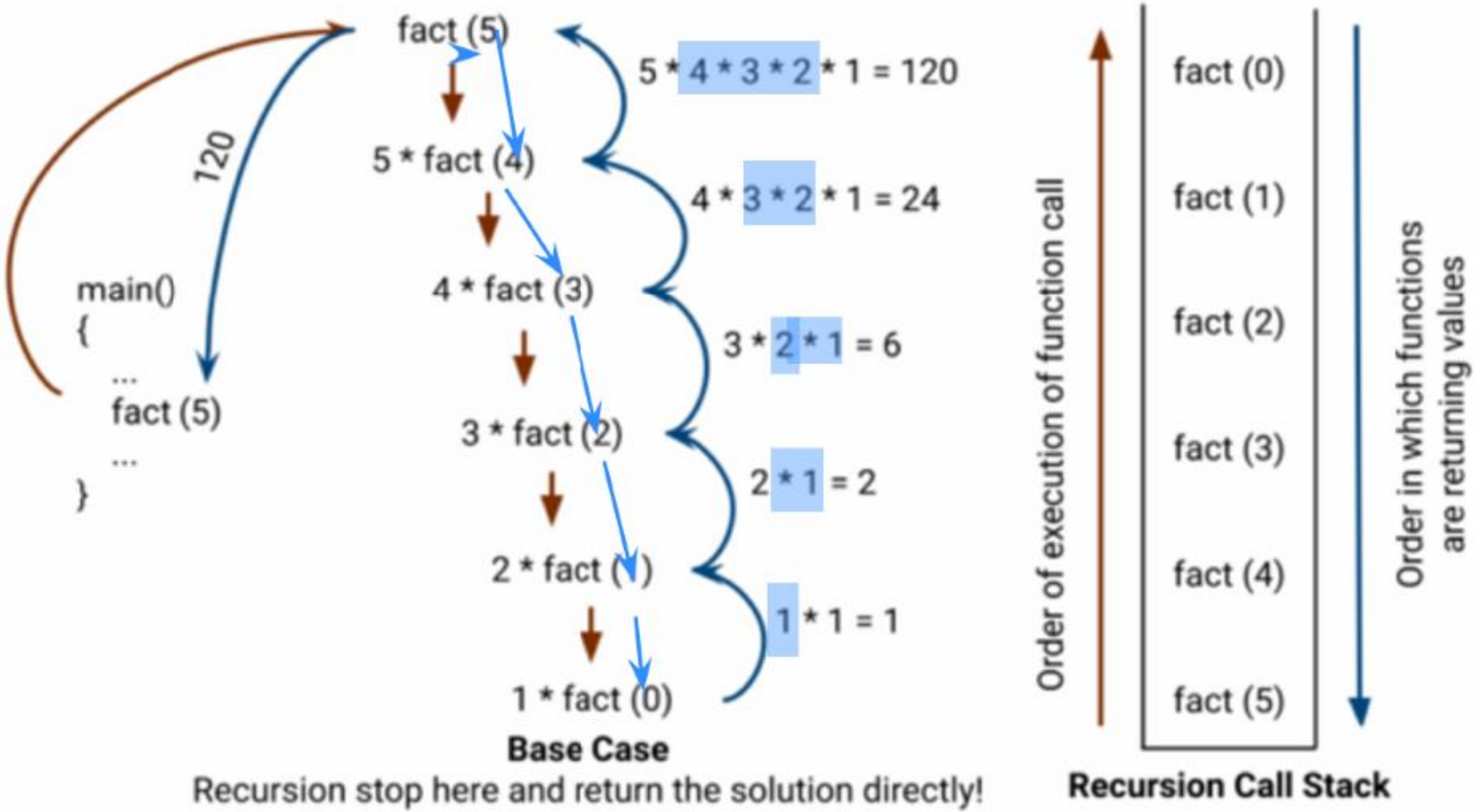
- Mathematical Equation:
- n if n == 0, n == 1;
- fib(n) = fib(n-1) + fib(n-2) otherwise;

- Recurrence Relation:
- T(n) = T(n-1) + T(n-2) + O(1)

- Recursive program:
- Input: n = 5
- Output:
- Fibonacci series of 5 numbers is : 0 1 1 2 3

How Data Structure Recursive function is implemented?



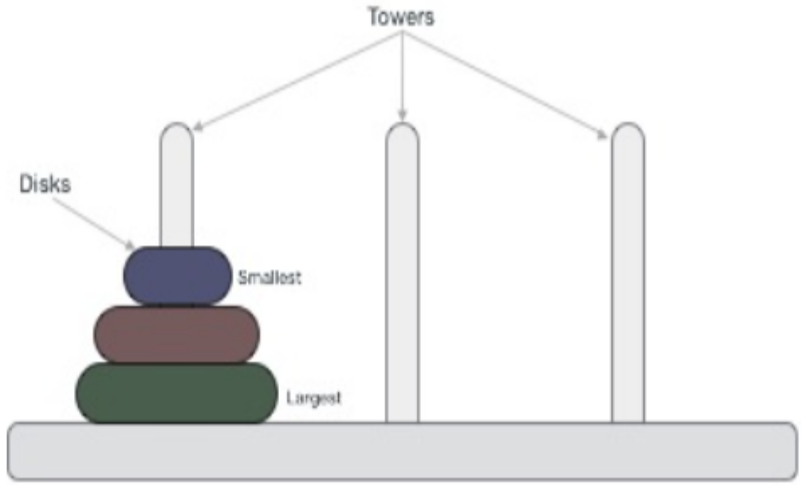

fact (5)
5 * 4 * 3 * 2 * 1 = 120
5 * fact (4)
4 * 3 * 2 * 1 = 24
4 * fact (3)
3 * 2 * 1 = 6
3 * fact (2)
2 * 1 = 2
2 * fact (1)
1 * 1 = 1
1 * fact (0)
120
main()
{
...
fact (5)
...
}
Base Case
Recursion stop here and return the solution directly!
Order of execution of function call
fact (0)
fact (1)
fact (2)
fact (3)
fact (4)
fact (5)
Order in which functions are returning values
Recursion Call Stack

What is Tower of Hanoi?

- A mathematical puzzle consisting of three towers and more than one ring is known as Tower of Hanoi.

- Tower of Hanoi

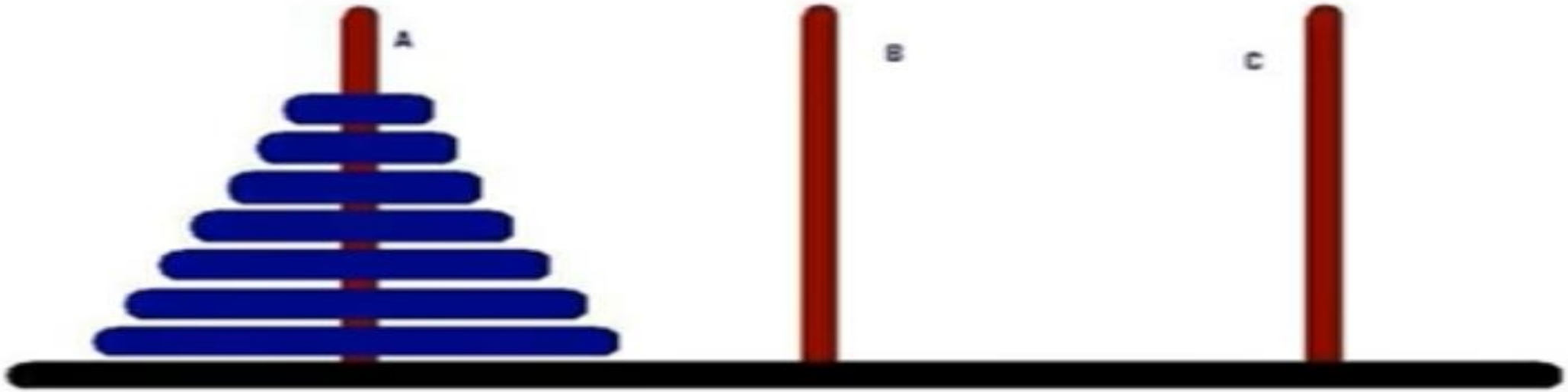
- The rings are of different sizes and are stacked in ascending order, i.e., the smaller one sits over the larger one. In some of the puzzles, the number of rings may increase, but the count of the tower remains the same.



Tower of Hanoi

Tower oh Hanoi

- The tower of Hanoi is mathematical puzzle.



- The objective of the puzzle is to move the entire stack to another rod.

What are the rules to be followed by Tower of Hanoi?

- **The Tower of Hanoi puzzle is solved by moving all the disks to another tower by not violating the sequence of the arrangements.**

The rules to be followed by the Tower of Hanoi are -

1. Only one disk can be moved among the towers at any given time.
2. Only the "top" disk can be removed.
3. No large disk can sit over a small disk.

Algorithm 1: Recursive algorithm for solving Towers of Hanoi

```
1 function recursiveHanoi(n, s, a, d)
2     if n == 1 then
3         print(s + " to " + d);
4         return;
5     end
6     recursiveHanoi(n − 1, s, d, a);
7     print(s + " to " + d);
8     recursiveHanoi(n − 1, a, s, d);
9 end
```

Home Work

- Implement Tower of Hanoi Program
- No of Disk=3
- No of Disk=5
- No of Disk=n