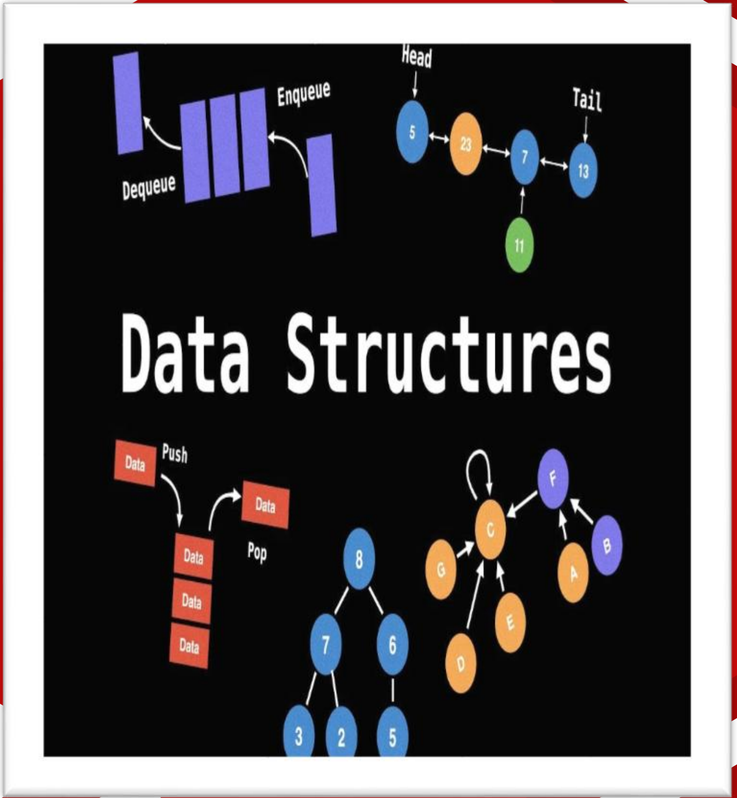
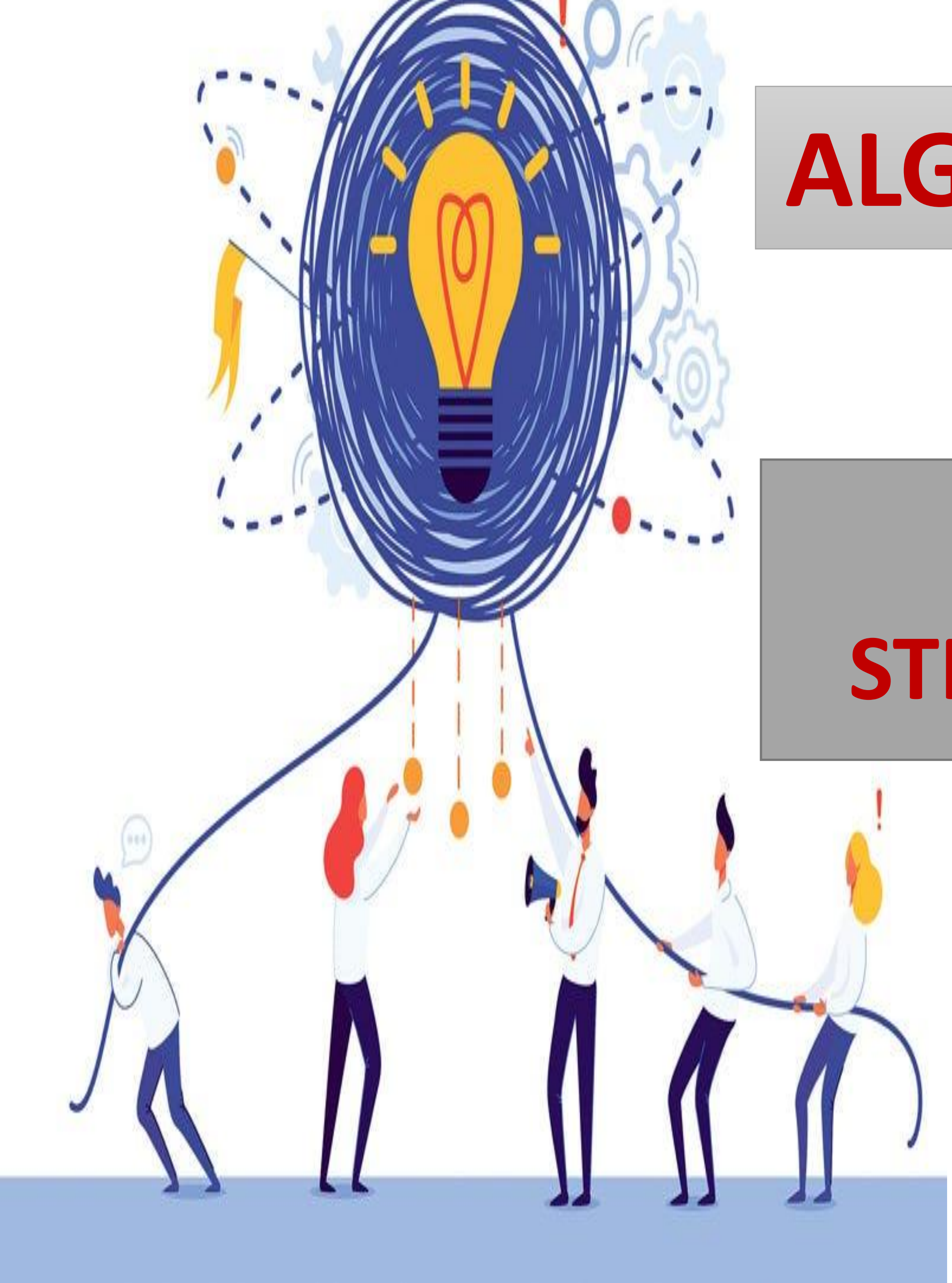
Algorithms and Data Structures

Introduction to Data Structures

Session : Day 1

Dr Kiran Waghmare
CDAC Mumbai





ALGORITHM

+

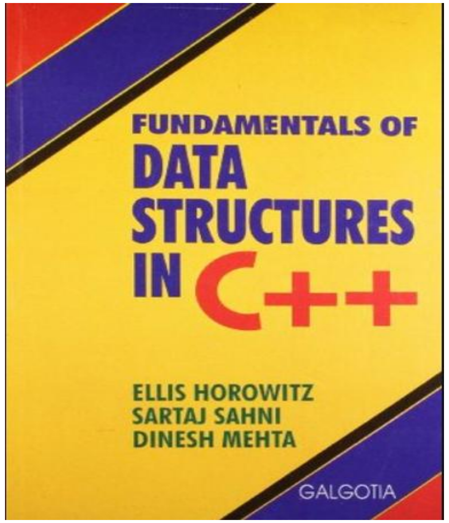
DATA STRUCTURE

Logical Real life Problem

1. Travelling from Mumbai to Goa
2. Criteria for Marriage
3. ATM money withdrawal
4. Online Money transfer
5. Online shopping

Module 2: Algorithms and Data Structures

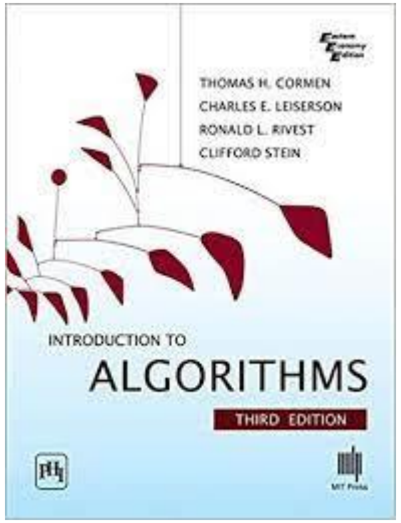
- **Text Book:**
  - Fundamentals of Data Structures in C++ by Horowitz, Sahani & Mehta
- **Topics:**
  - 1.Problem Solving & Computational Thinking
  - 2.Introduction to Data Structures & Recursion
  - 3.Stacks
  - 4.Queues
  - 5.Linked List Data Structures
  - 6.Trees & Applications
  - 7.Introduction to Algorithms
  - 8.Searching and Sorting
  - 9.Hash Functions and Hash Tables
  - 10.Graph & Applications
  - 11.Algorithm Designs



FUNDAMENTALS OF DATA STRUCTURES IN C++

ELLIS HOROWITZ
SARTAJ SAHNI
DINESH MEHTA

GALGOTIA

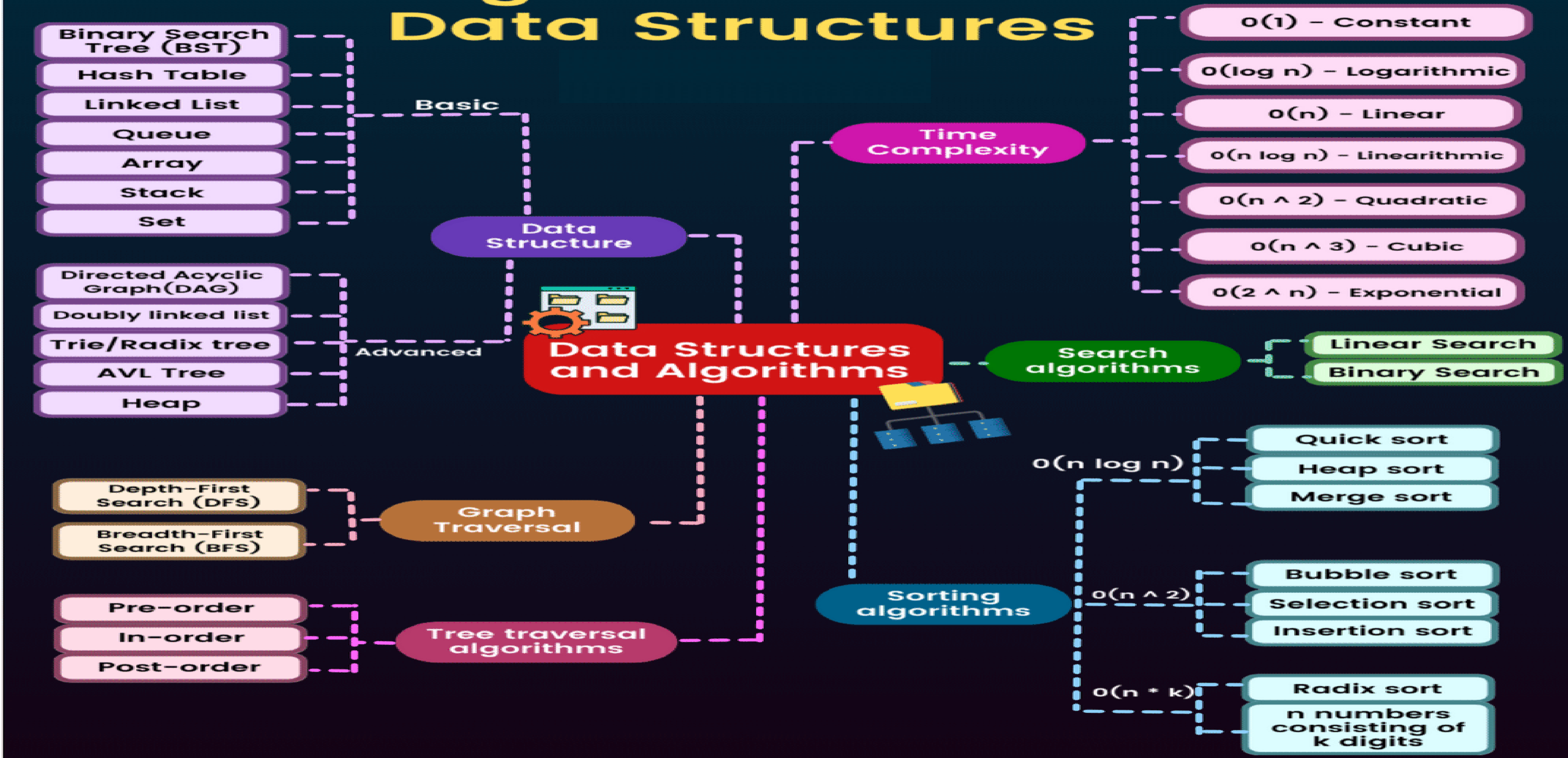


THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN

INTRODUCTION TO ALGORITHMS

THIRD EDITION

Algorithms and Data Structures



- Data Structures and Algorithms
  - Data Structure
    - Basic
      - Binary Search Tree (BST)
      - Hash Table
      - Linked List
      - Queue
      - Array
      - Stack
      - Set
    - Advanced
      - Directed Acyclic Graph(DAG)
      - Doubly linked list
      - Trie/Radix tree
      - AVL Tree
      - Heap
  - Time Complexity
    - O(1) - Constant
    - O(log n) - Logarithmic
    - O(n) - Linear
    - O(n log n) - Linearithmic
    - O(n ^ 2) - Quadratic
    - O(n ^ 3) - Cubic
    - O(2 ^ n) - Exponential
  - Search algorithms
    - Linear Search
    - Binary Search
  - Sorting algorithms
    - O(n log n)
      - Quick sort
      - Heap sort
      - Merge sort
    - O(n ^ 2)
      - Bubble sort
      - Selection sort
      - Insertion sort
    - O(n * k)
      - Radix sort
      - n numbers consisting of k digits
  - Graph Traversal
    - Depth-First Search (DFS)
    - Breadth-First Search (BFS)
  - Tree traversal algorithms
    - Pre-order
    - In-order
    - Post-order

Agenda

- **Problem Solving & Computational Thinking**
- **Algorithm & Data Structure**
    - OODesign: ADTs
- **Recursion**
    - Base condition
    - Direct & indirect recursion
    - Memory allocation
    - Pros and Cons
    - Complexity analysis

Computational Thinking : Researcher

- Niklaus Wirth

Linus Torvalds

Martin Karplus, Michael Levitt, and Arieh Warshel

Why Study Algorithms and Data Structures?

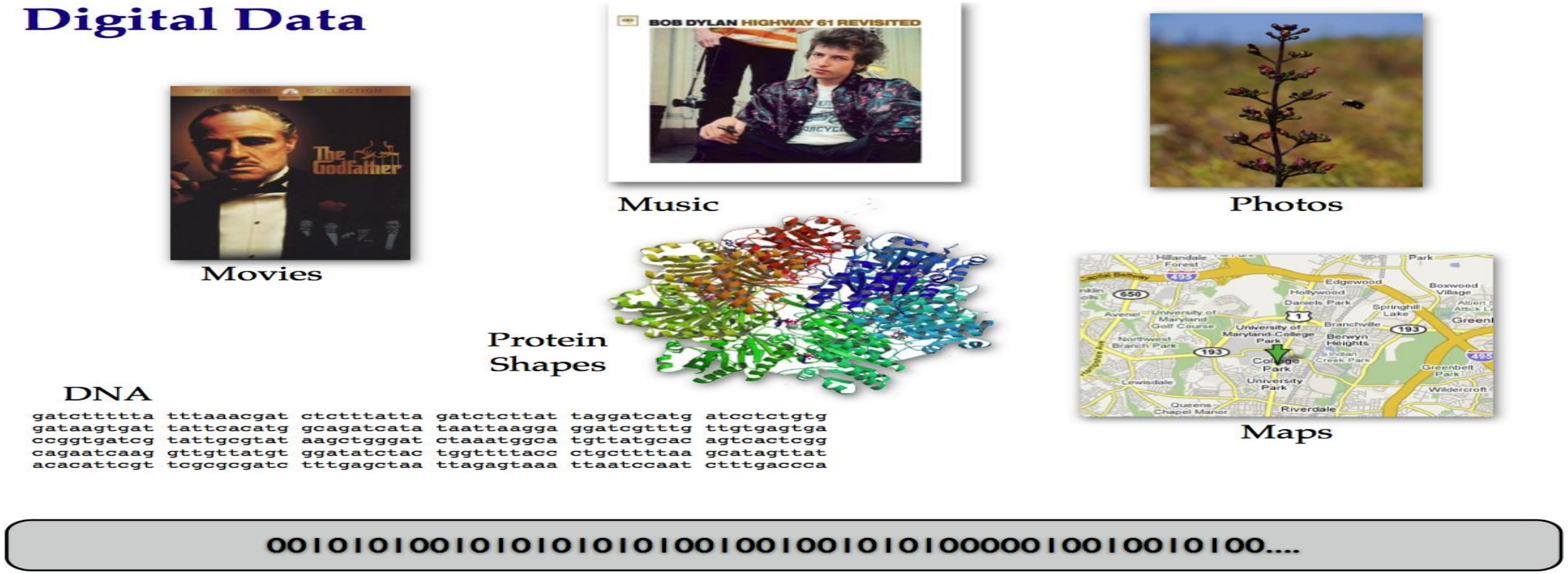
- World domination

Algorithms are Everywhere

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

Modern World of Computing

- Age of Big Data, birth of Data Science
- Digitization, communication, sensing, imaging…
- Entertainment, science, maps, health, environmental, banking…

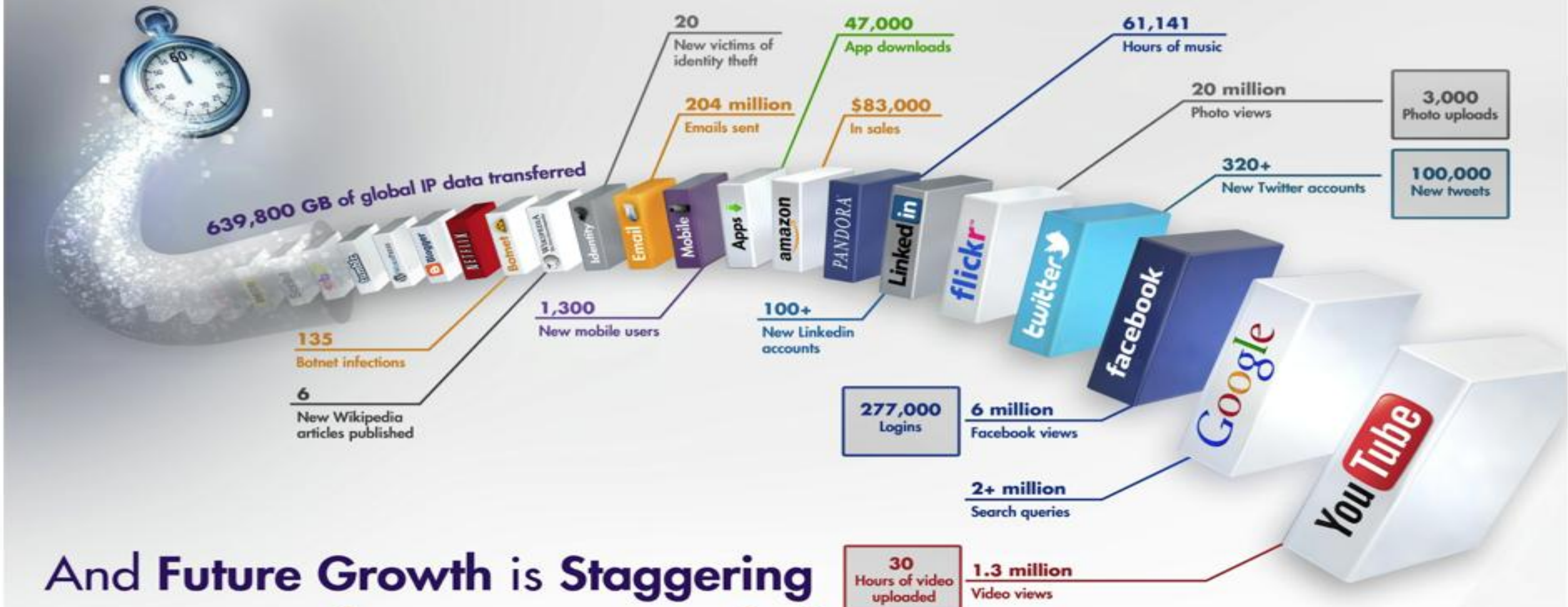


```
gatcttttta tttaaacgat ctctttatta gatctcttat taggatcatg atcctctgtg
gataagtgat tattcacatg gcagatcata taattaagga ggatcgtttg ttgtgagtga
ccggtgatcg tattgcgtat aagctgggat ctaaatggca tgttatgcac agtcactcgg
cagaatcaag gttgttatgt ggatatctac tggttttacc ctgcttttaa gcatagttat
acacattcgt tcgcgcgatc tttgagctaa ttagagtaaa ttaatccaat ctttgaccca
```

0010101001010101010100100100101010000010010010100....

- Volume, variety, velocity, variability
- What all happens in 1 Internet minute?

What Happens in an **Internet Minute?**



639,800 GB of global IP data transferred

135
Botnet infections

6
New Wikipedia articles published

20
New victims of identity theft

204 million
Emails sent

1,300
New mobile users

47,000
App downloads

$83,000
In sales

100+
New Linkedin accounts

61,141
Hours of music

20 million
Photo views

3,000
Photo uploads

320+
New Twitter accounts

100,000
New tweets

277,000
Logins

6 million
Facebook views

2+ million
Search queries

30
Hours of video uploaded

1.3 million
Video views

And **Future Growth** is **Staggering**

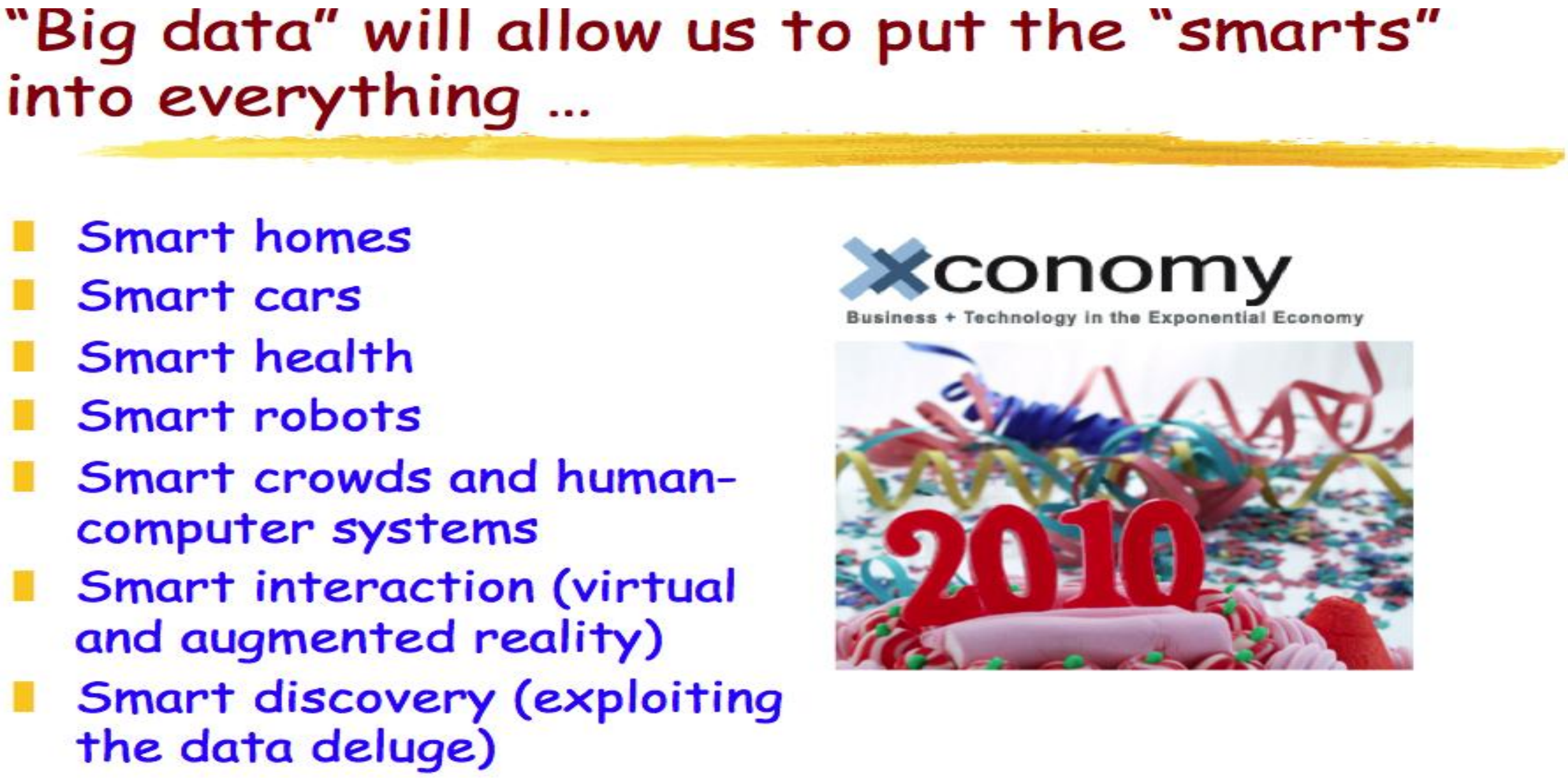
Today, the number of networked devices = the global population

By **2015**, the number of networked devices = 2x the global population

In **2015**, it would take you **5 years** to view all video crossing IP networks each **second**

Credit: Intel Corporation

Intelligent Computational Systems



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

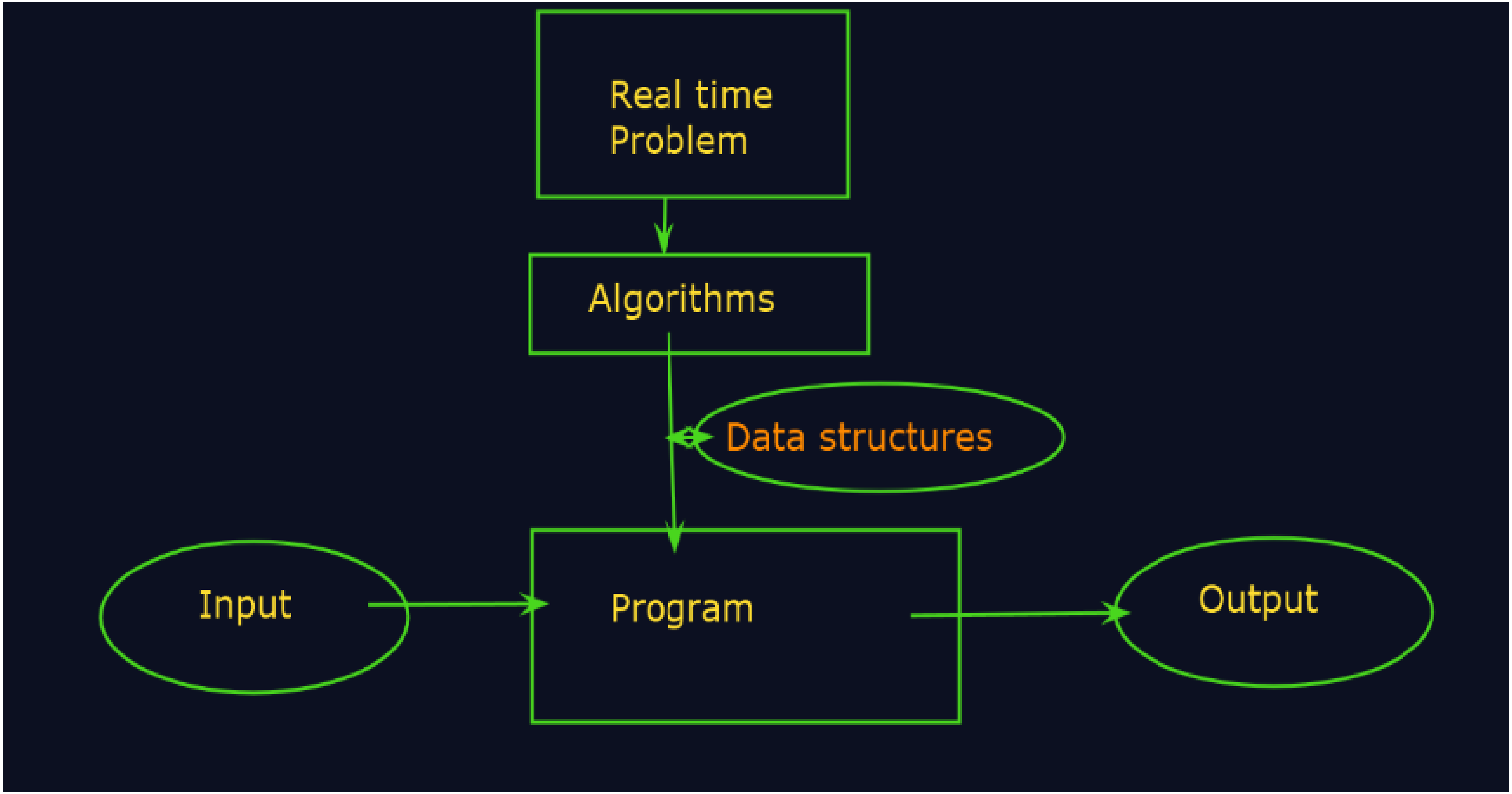
- Smart homes
- Smart cars
- Smart health
- Smart robots
- Smart crowds and human-computer systems
- Smart interaction (virtual and augmented reality)
- Smart discovery (exploiting the data deluge)

Xconomy

Business + Technology in the Exponential Economy

Definition

- **Data:**
  - Collection of Raw facts.
- **Algorithm:**
  - Outline, the essence of a computational procedure, step-by-step instructions.
- **Program:**
  - An implementation of an algorithm in some programming language
- **Data Structure:**
  - Organization of data needed to solve the problem.
  - The programmatic way of storing data so that data can be used efficiently


Real time Problem
Algorithms
Data structures
Input
Program
Output

Algorithm

- **An *algorithm* is a sequence of unambiguous instructions/operations for solving a problem, for obtaining a required output for any legitimate input in a finite amount of time.**

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

Analysis of Algorithms

- **How good is the algorithm?**
  - Correctness
  - Time efficiency
  - Space efficiency

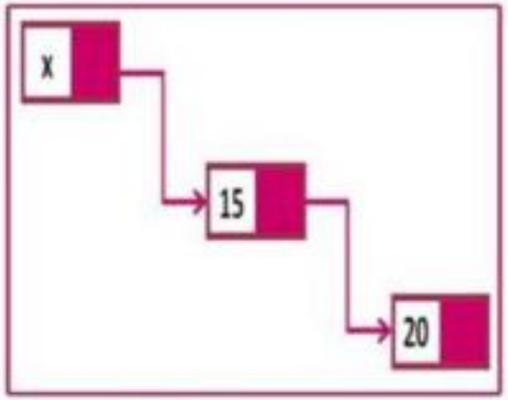
- **Does there exist a better algorithm?**
  - Lower bounds
  - Optimality

Analysis of Algorithms

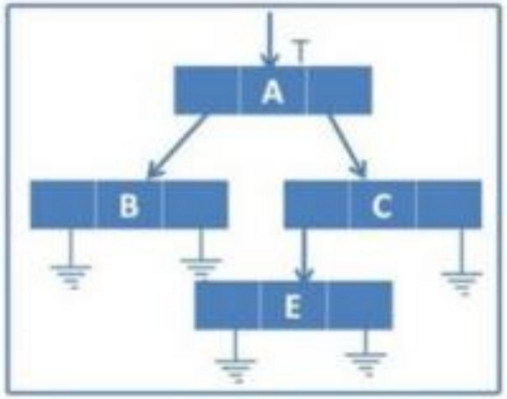
- An algorithm is said to be efficient and fast, if **it takes less time to execute and consumes less memory space**.
- The performance of an algorithm is measured on the basis of following properties :

1. Time Complexity
2. Space Complexity

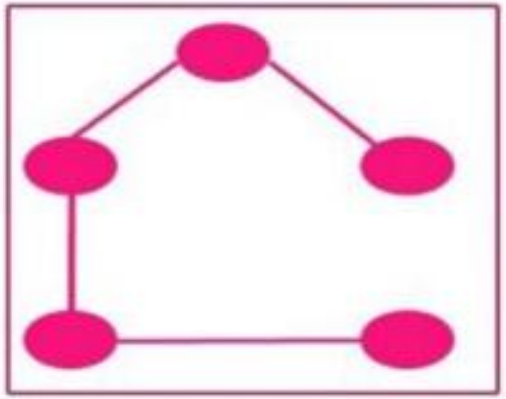
Sorting



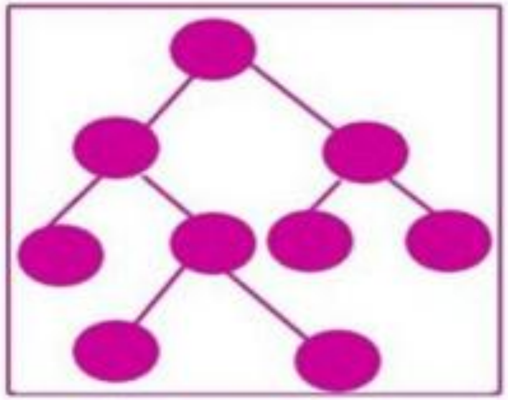
Link list



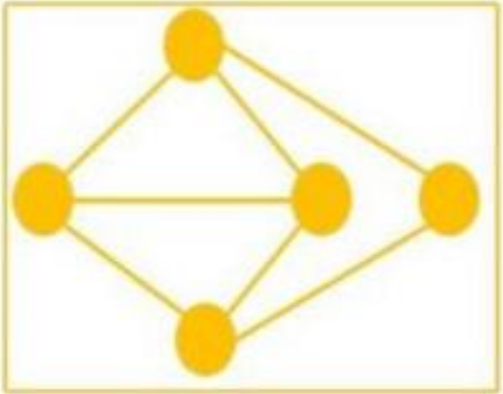
list



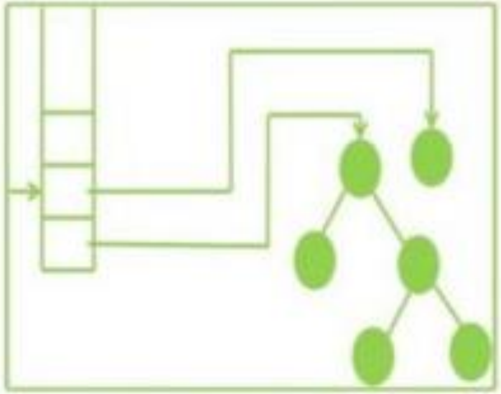
spanning tree



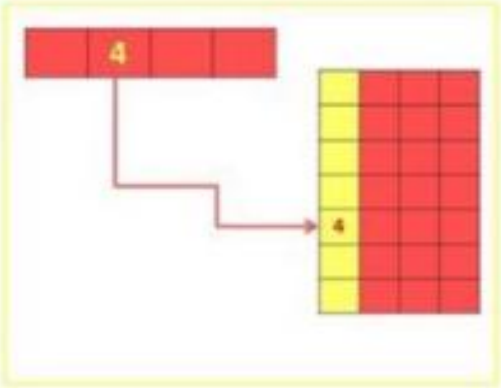
Tree



Graph

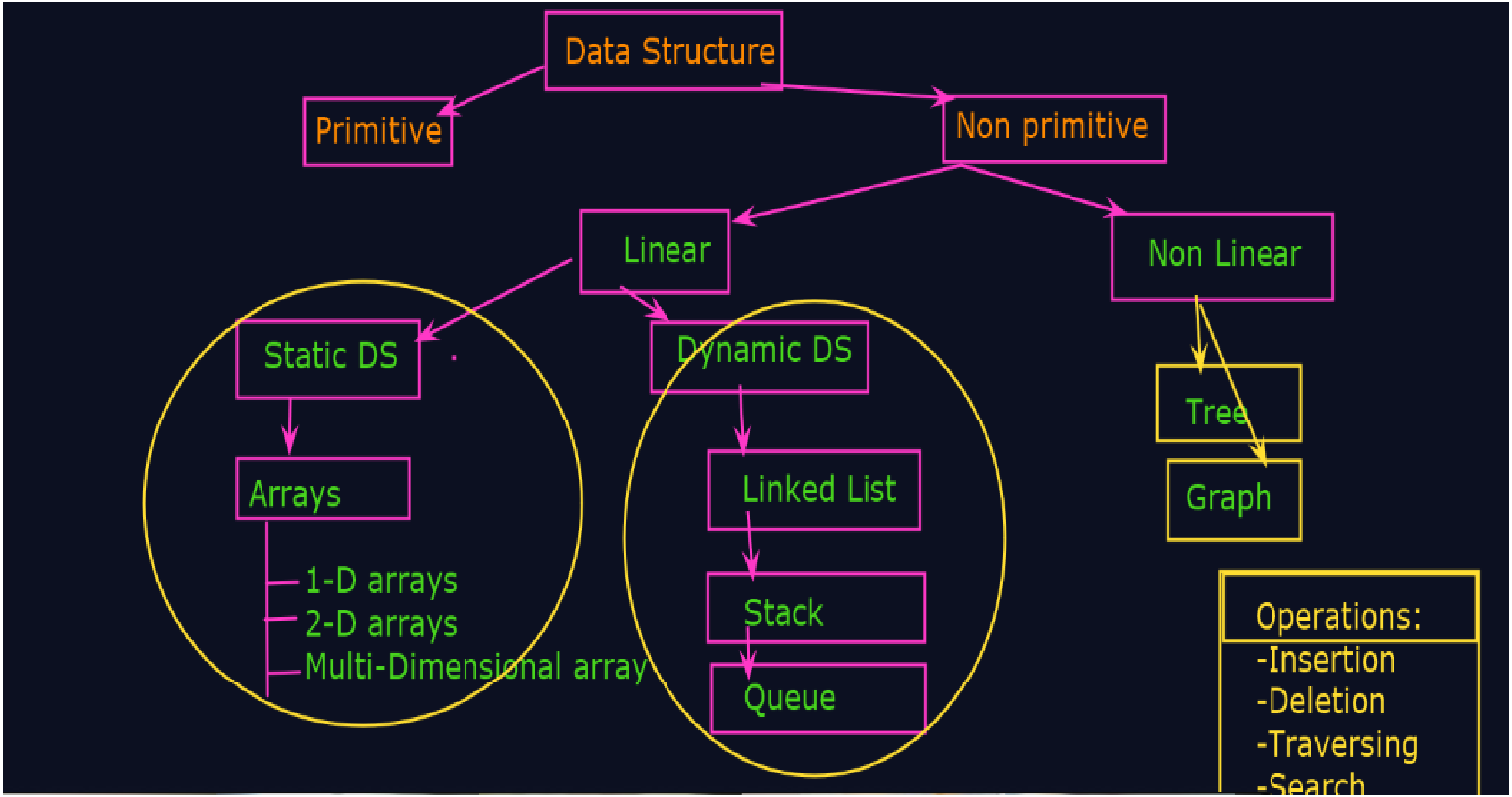


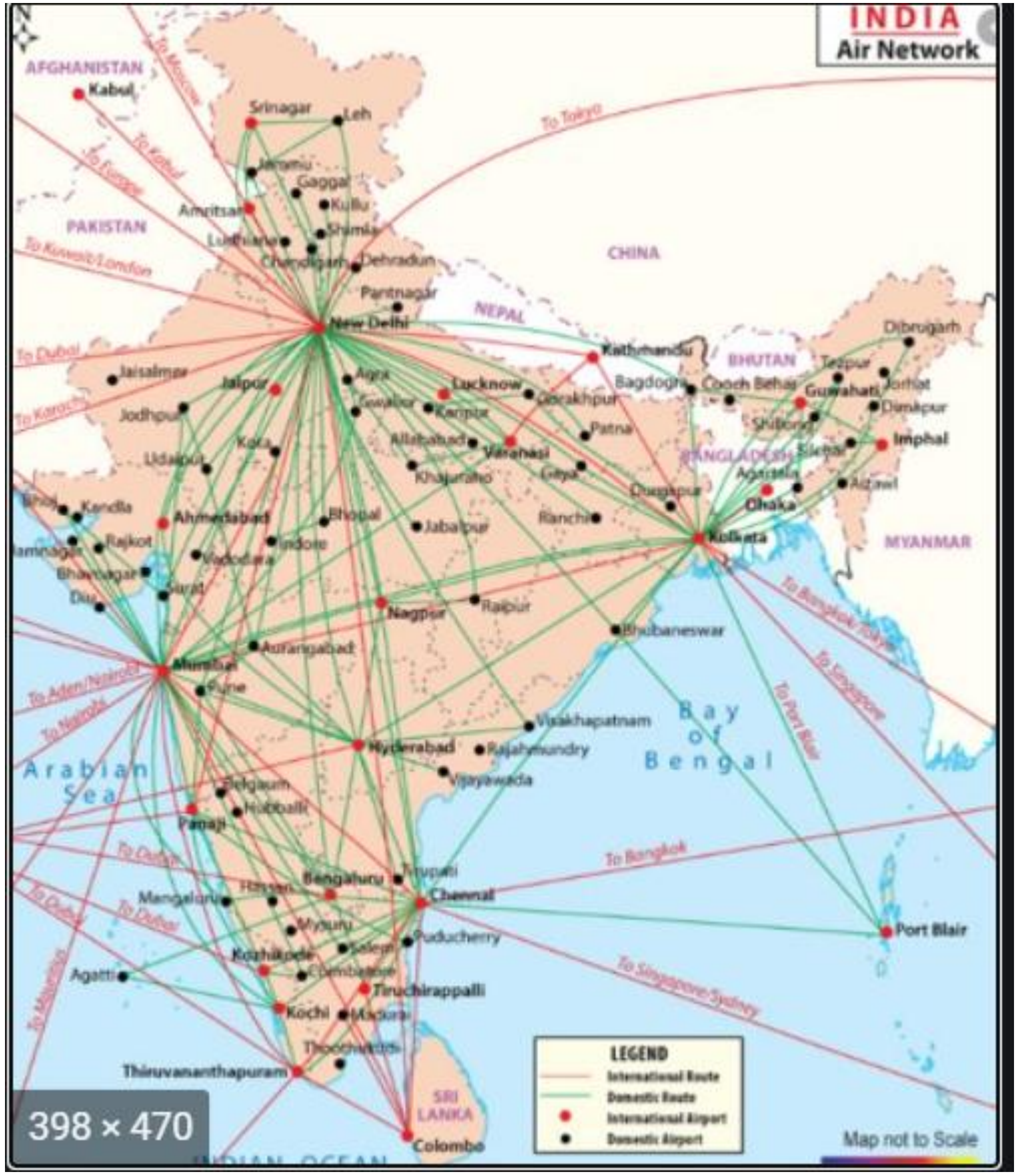
Stack

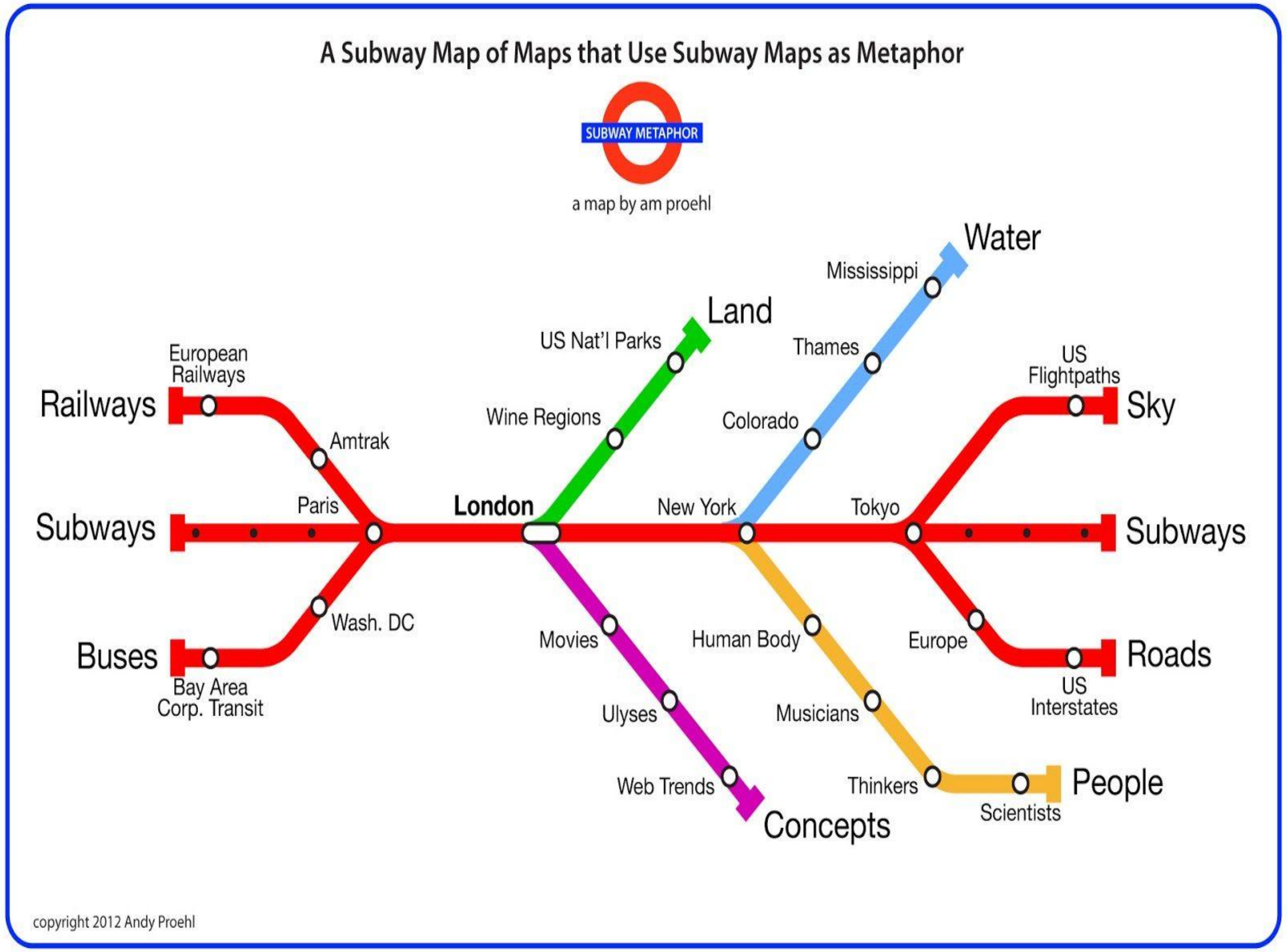


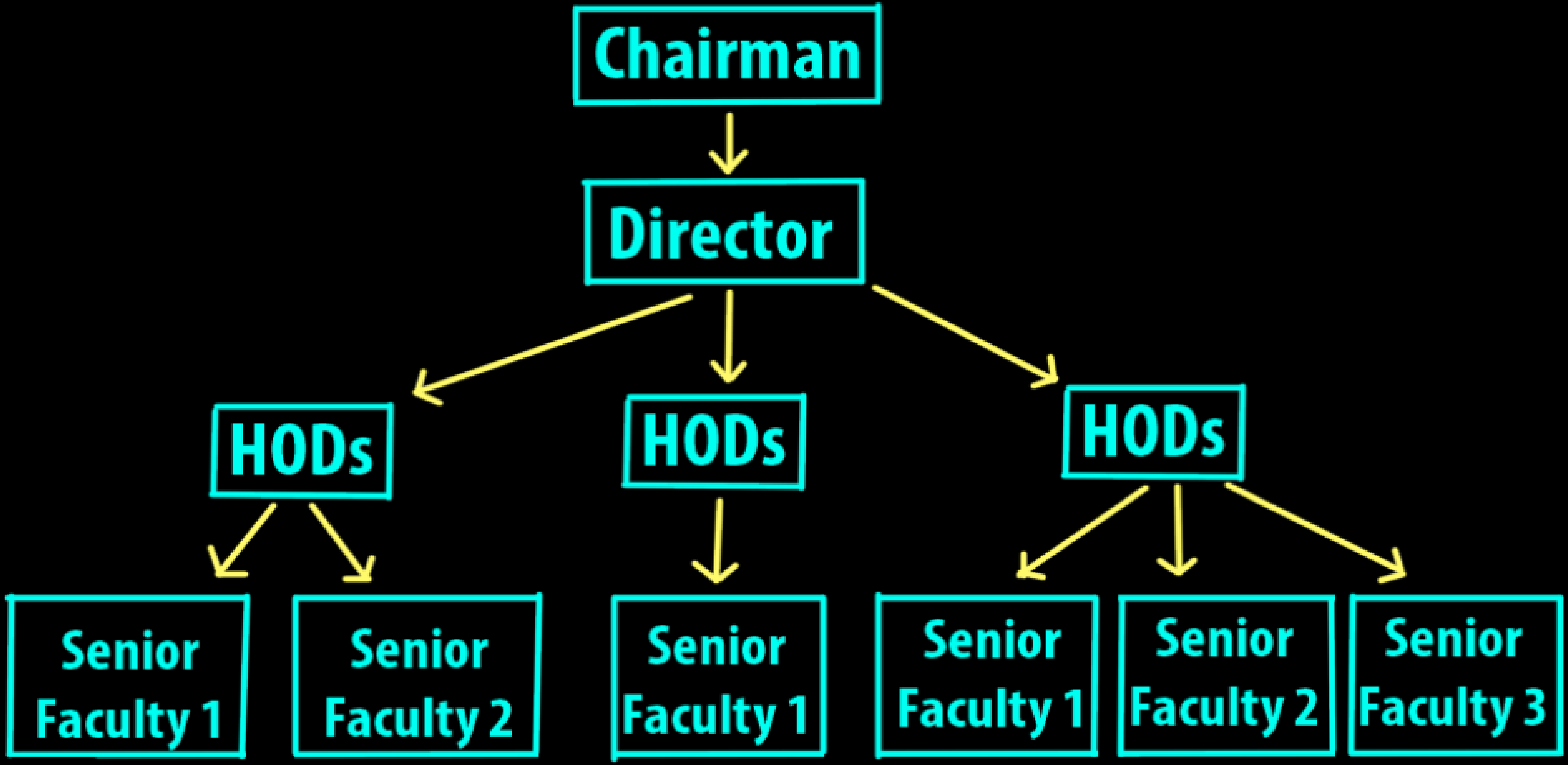
Hashing

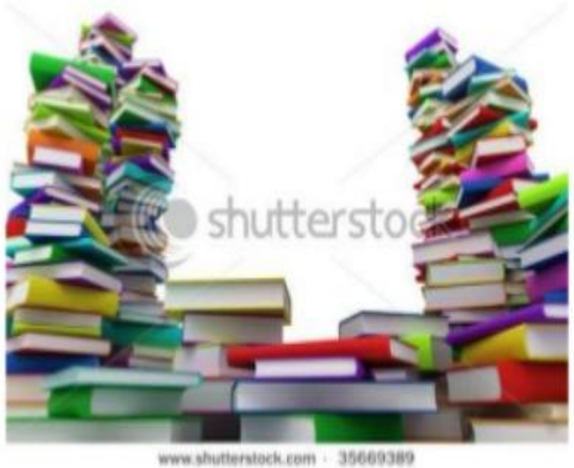
By...navinkumardhoprephotography.com

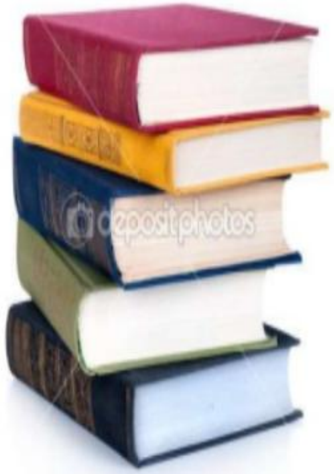

Data Structure
Primitive
Non primitive
Linear
Non Linear
Static DS
Dynamic DS
Tree
Graph
Arrays
1-D arrays
2-D arrays
Multi-Dimensional array
Linked List
Stack
Queue
Operations:
-Insertion
-Deletion
-Traversing
-Search


INDIA
Air Network
AFGHANISTAN
Kabul
PAKISTAN
CHINA
NEPAL
BHUTAN
BANGLADESH
MYANMAR
SRI LANKA
Arabian Sea
Bay of Bengal
INDIAN OCEAN
Srinagar
Leh
Jammu
Amritsar
Ludhiana
Chandigarh
Dehradun
Pantnagar
New Delhi
Kathmandu
Jaisalmer
Jaipur
Agra
Gwalior
Lucknow
Kanpur
Gorakhpur
Allahabad
Varanasi
Patna
Bagdogra
Cooch Behar
Guwahati
Tezpur
Dibrugarh
Jorhat
Dimapur
Imphal
Shillong
Silchar
Agartala
Aizawl
Dhaka
Jodhpur
Udaipur
Kota
Khajuraho
Gaya
Durgapur
Ranchi
Kolkata
Kandla
Rajkot
Jamnagar
Bhavnagar
Diu
Ahmedabad
Vadodara
Surat
Bhopal
Indore
Jabalpur
Nagpur
Raipur
Bhubaneswar
Mumbai
Aurangabad
Pune
Visakhapatnam
Hyderabad
Rajahmundry
Vijayawada
Belgaum
Hubballi
Panaji
Mangaluru
Hassan
Bengaluru
Tirupati
Chennai
Mysuru
Salem
Puducherry
Kozhikode
Coimbatore
Agatti
Kochi
Tiruchirappalli
Madurai
Thoothukudi
Thiruvananthapuram
Colombo
Port Blair
To Moscow
To Tokyo
To Kabul
To Europe
To Kuwait/London
To Dubai
To Karachi
To Aden/Nairobi
To Nairobi
To Dubai
To Dubai
To Dubai
To Mauritius
To Bangkok Tokyo
To Singapore
To Port Blair
To Bangkok
To Singapore Sydney
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
Water
Mississippi
Land
Thames
US Nat'l Parks
European Railways
Railways
Wine Regions
Colorado
US Flightpaths
Sky
Amtrak
Paris
London
New York
Tokyo
Subways
Subways
Wash. DC
Movies
Human Body
Europe
Buses
Roads
Bay Area Corp. Transit
Ulyses
Musicians
US Interstates
Web Trends
Thinkers
People
Scientists
Concepts
copyright 2012 Andy Proehl


Chairman
Director
HODs
HODs
HODs
Senior Faculty 1
Senior Faculty 2
Senior Faculty 1
Senior Faculty 1
Senior Faculty 2
Senior Faculty 3


shutterstock
www.shutterstock.com · 35669389


depositphotos

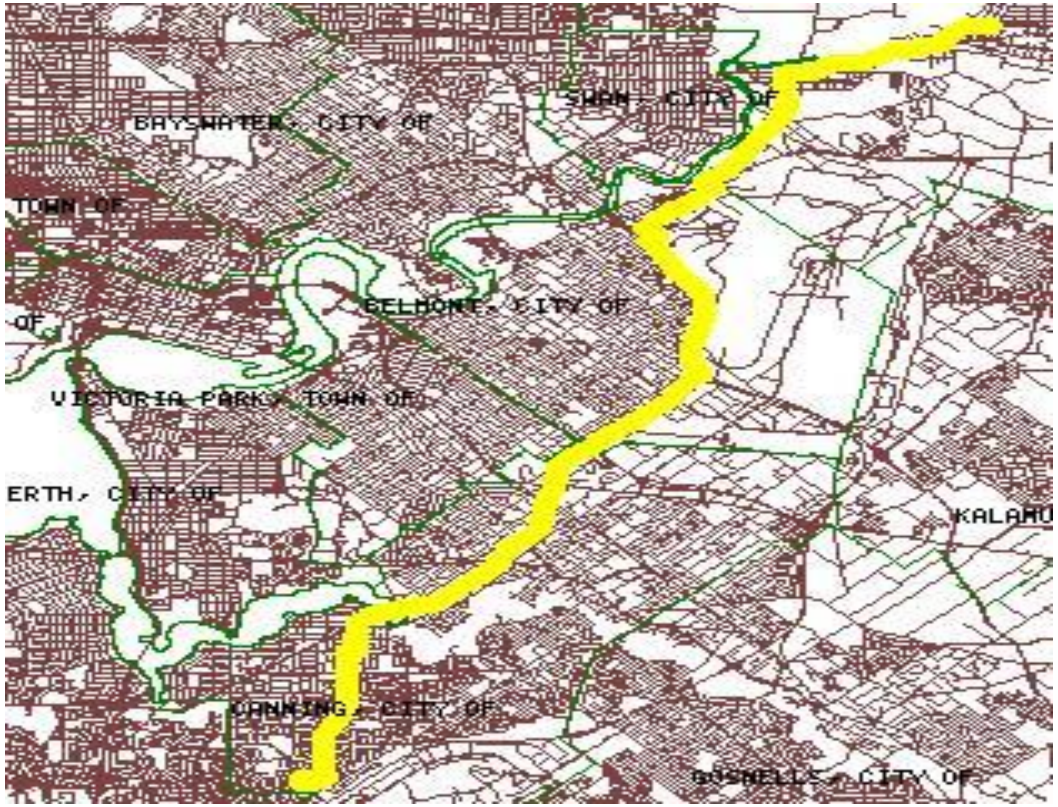
shutterstock
www.shutterstock.com · 10903984

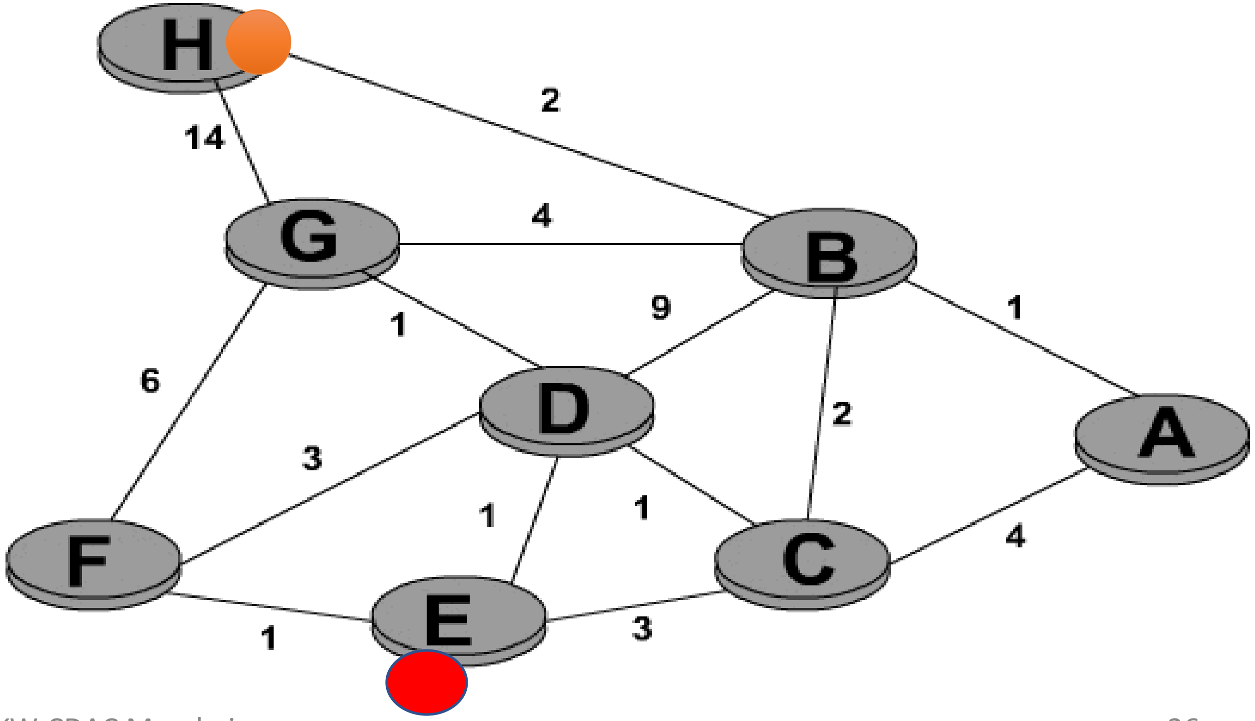
Aaaaaaarrrrrggghh!!!!
?

LAST IN LAST OUT
ONE WAY
FIRST IN FIRST OUT

Why you want to study Algorithms?

- Simply to be cool to invent something in computer science
- Example: Shortest Path Problem and Algorithm
- Used in GPS and Mapquest or Google Maps

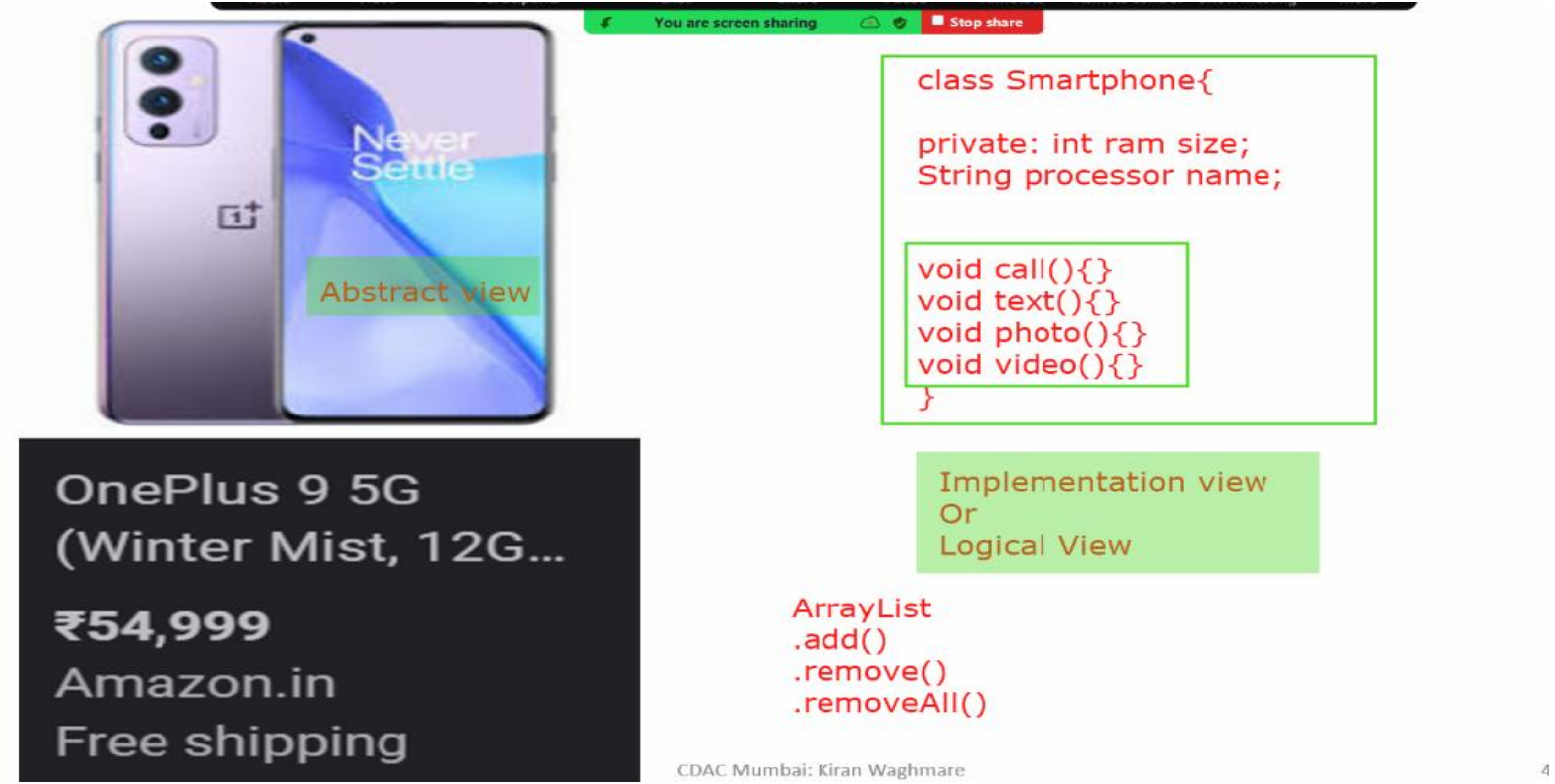




Abstract Data Type (ADT)

ADT

- Abstract Data type (ADT) is a type (or class) for objects whose behaviour is defined by a set of value and a set of operations.
- The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.
- It does not specify how data will be organized in memory and what algorithms will be used for implementing the operations.
- It is called "abstract" because it gives an implementation-independent view.
- The process of providing only the essentials and hiding the details is known as abstraction.
- All primitive data types support basic operations,+,-,*,/ etc



Abstract view

OnePlus 9 5G
(Winter Mist, 12G...

₹54,999

Amazon.in

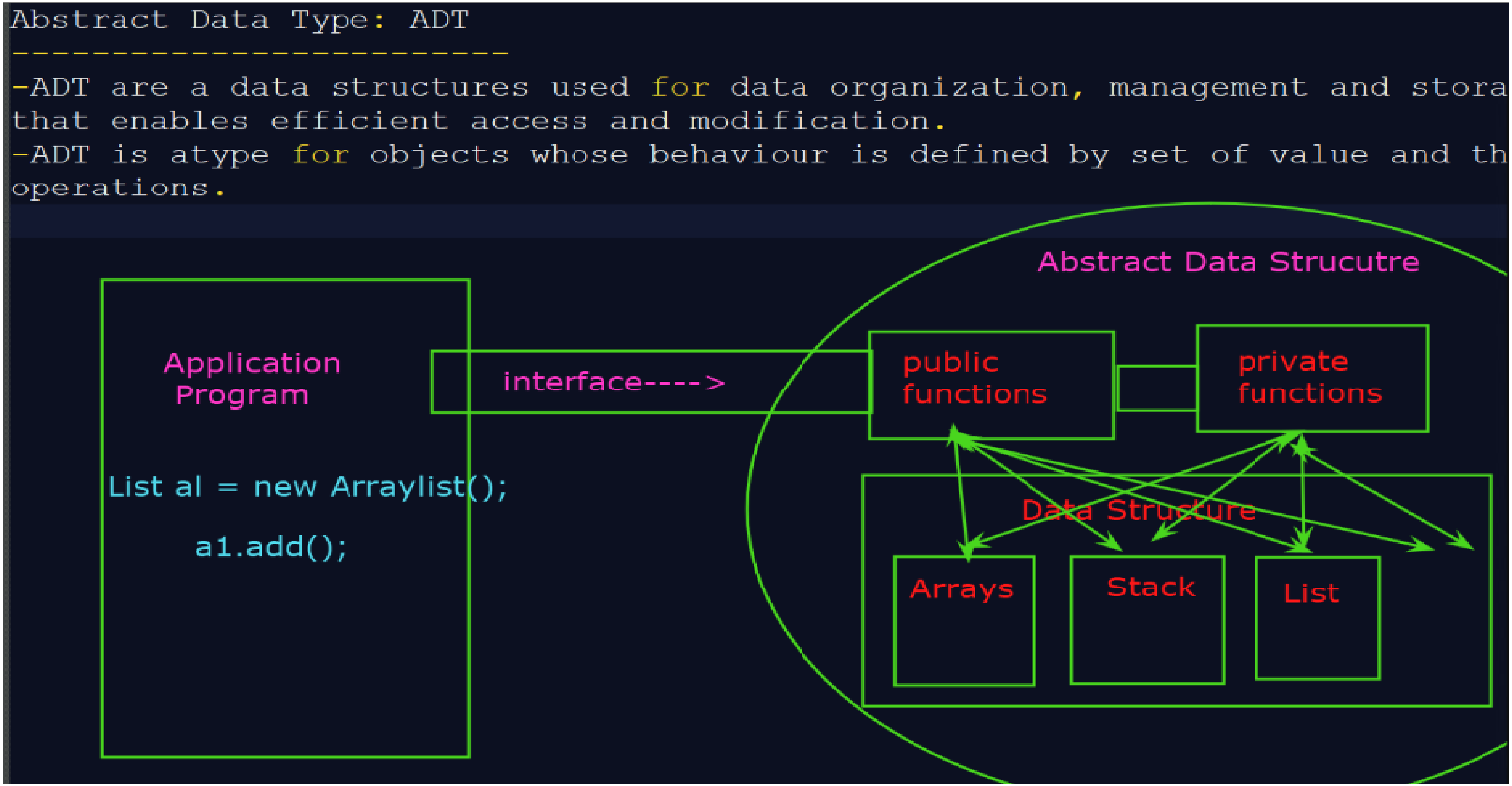
Free shipping

```
class Smartphone{

private: int ram size;
String processor name;


void call(){}
void text(){}
void photo(){}
void video(){}
}
```

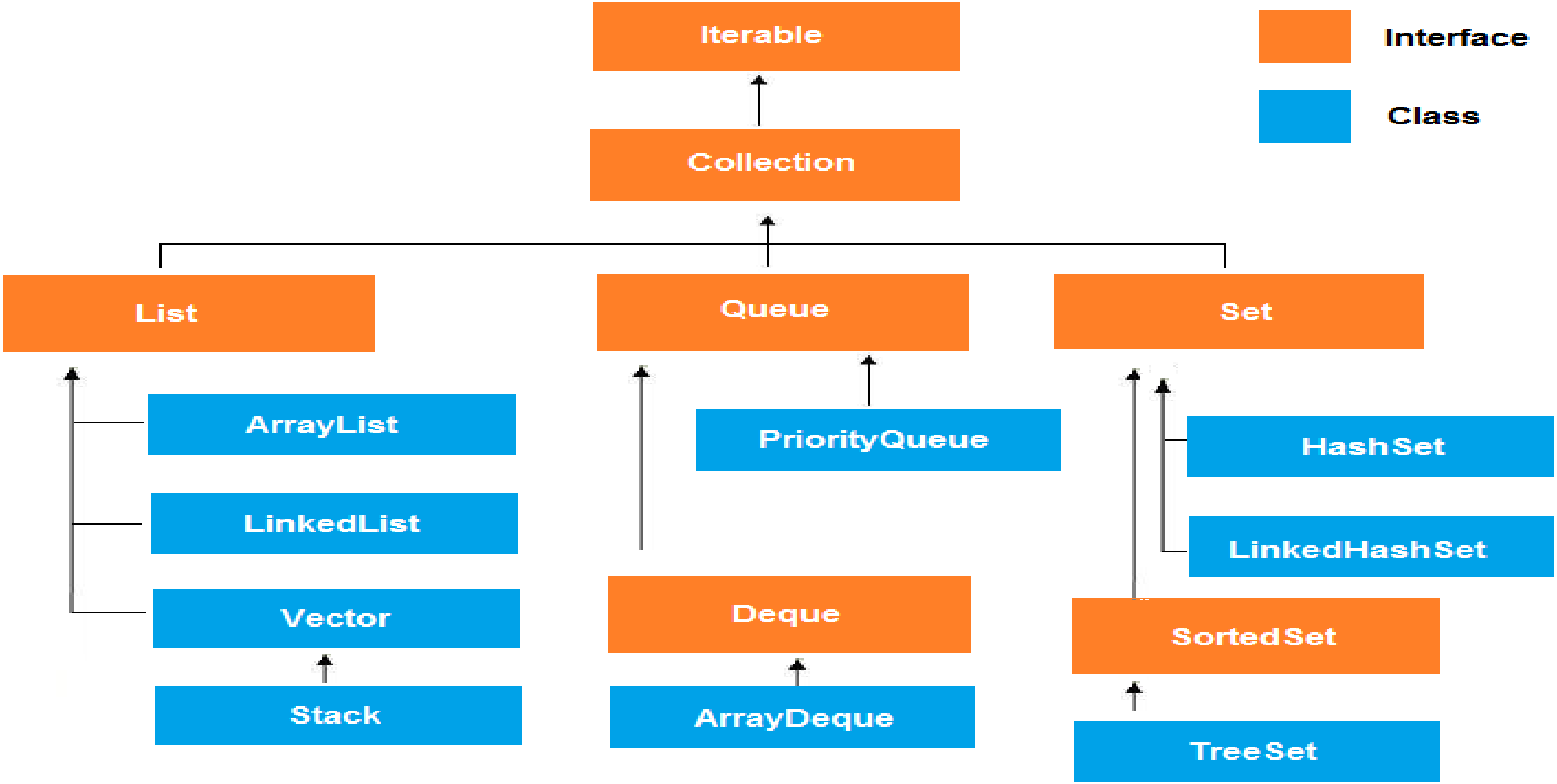
Implementation view
Or
Logical View

```
ArrayList
.add()
.remove()
.removeAll()
```

CDAC Mumbai: Kiran Waghmare 4



```
Abstract Data Type: ADT
---------------------------
-ADT are a data structures used for data organization, management and stora
that enables efficient access and modification.
-ADT is atype for objects whose behaviour is defined by set of value and th
operations.
```



Iterable

Collection

List

Queue

Set

ArrayList

LinkedList

Vector

Stack

PriorityQueue

Deque

ArrayDeque

HashSet

LinkedHashSet

SortedSet

TreeSet

🟧 Interface

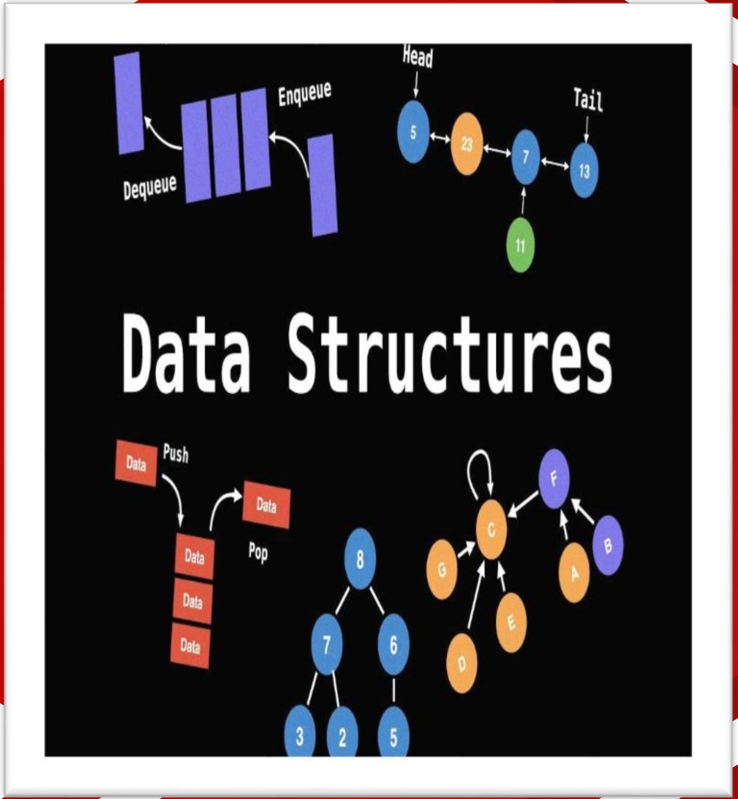
🟦 Class

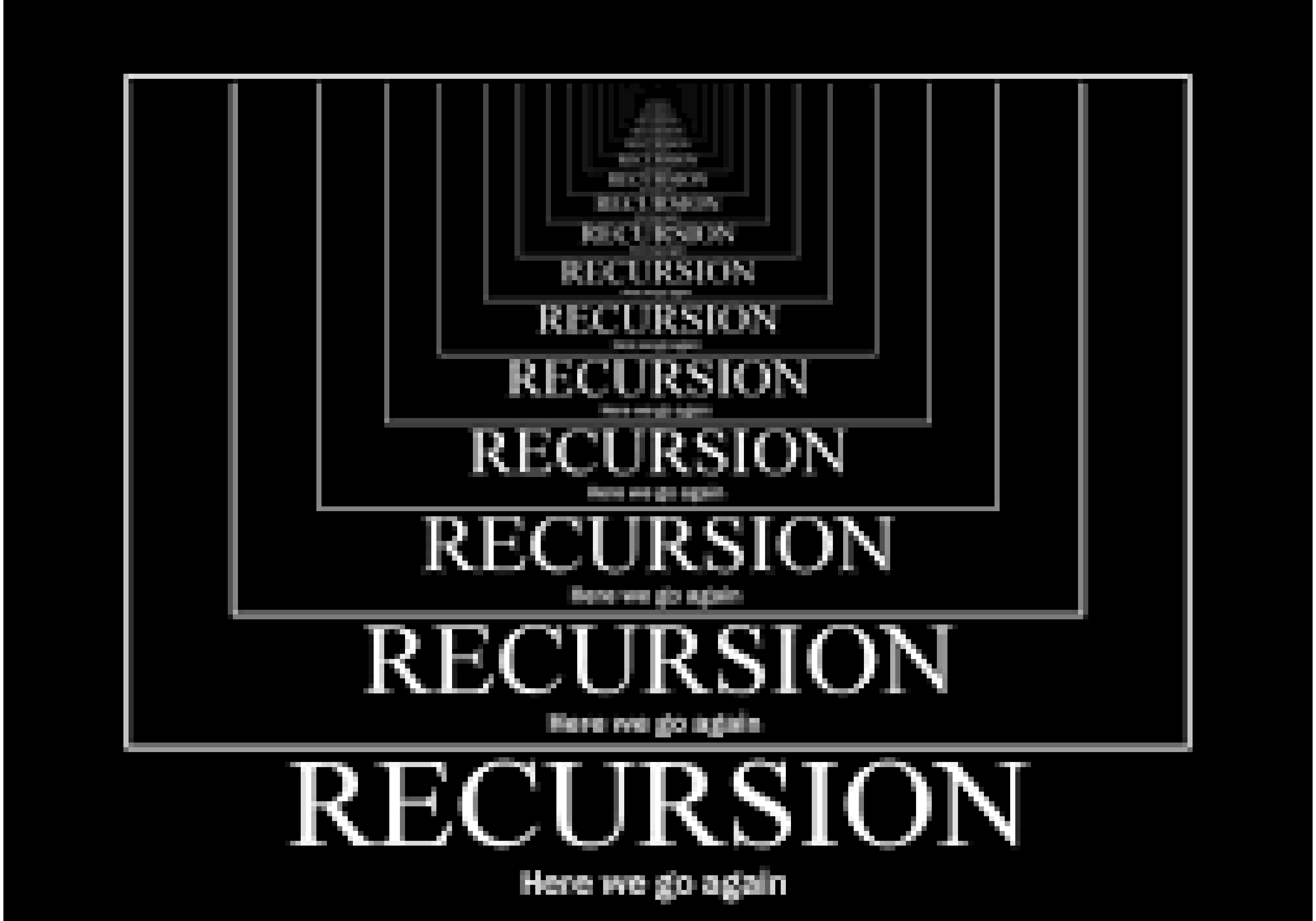
Algorithms and Data Structures

Recursion

Session : Day 1

Dr Kiran Waghmare
CDAC Mumbai

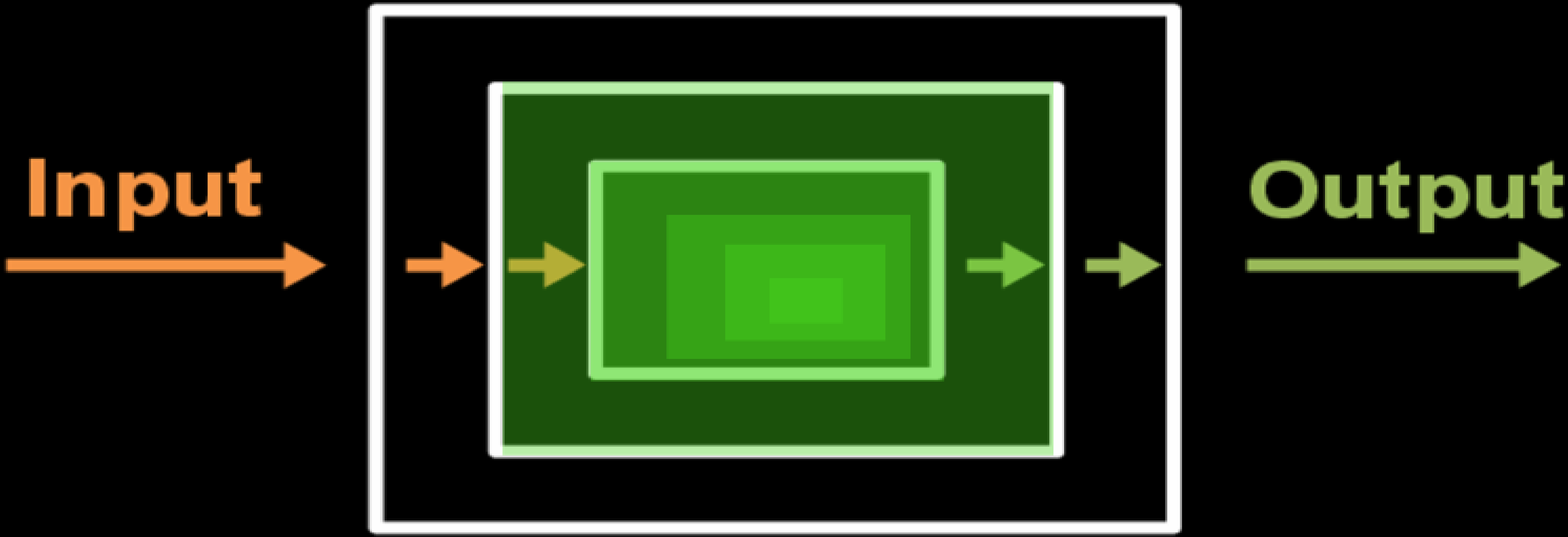



RECURSION
Here we go again
RECURSION
Here we go again
RECURSION
Here we go again
RECURSION
Here we go again
RECURSION
Here we go again
RECURSION
RECURSION
RECURSION

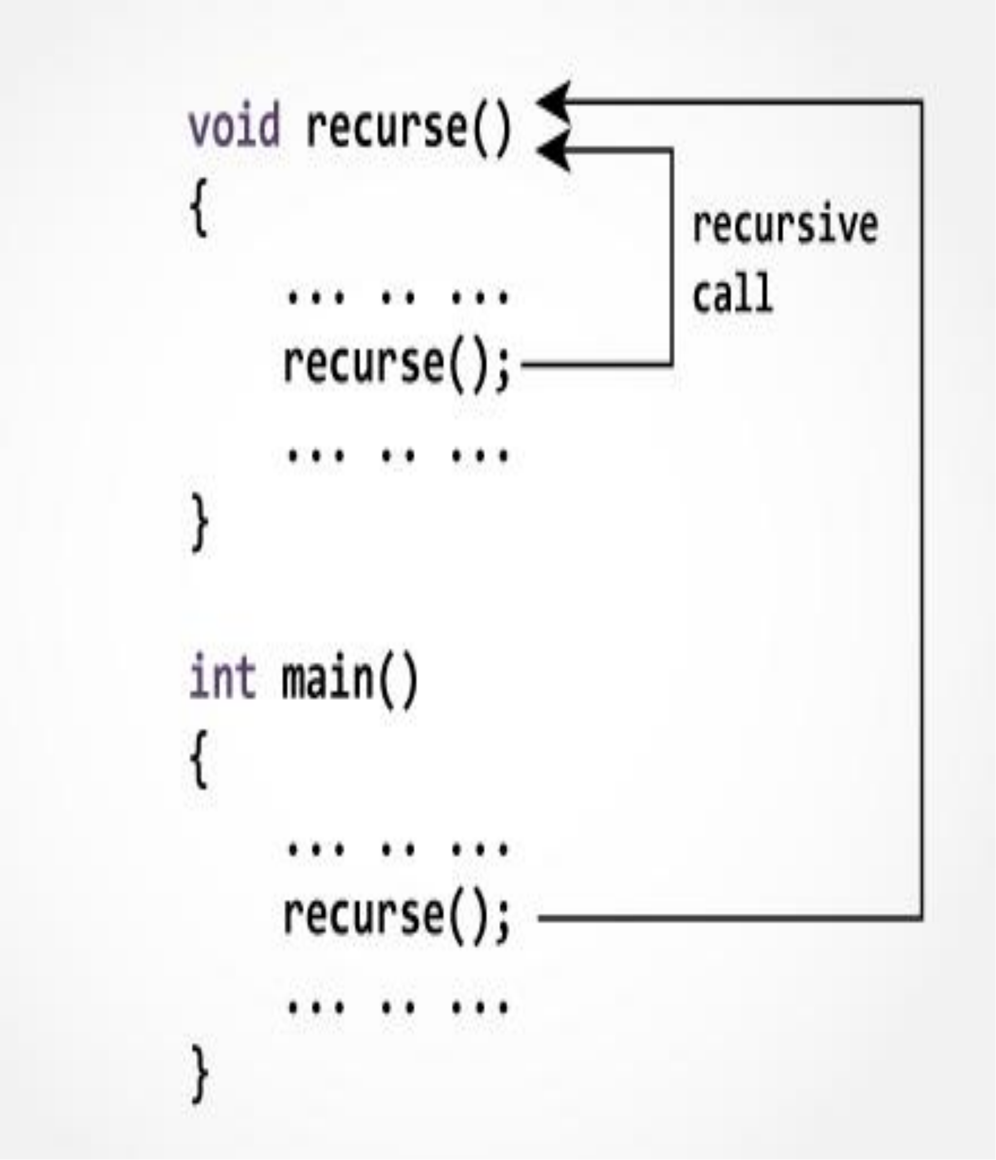
Topics

1. Recursive definitions and Processes
2. Writing Recursive Programs
3. Efficiency in Recursion
4. Towers of Hanoi problem.

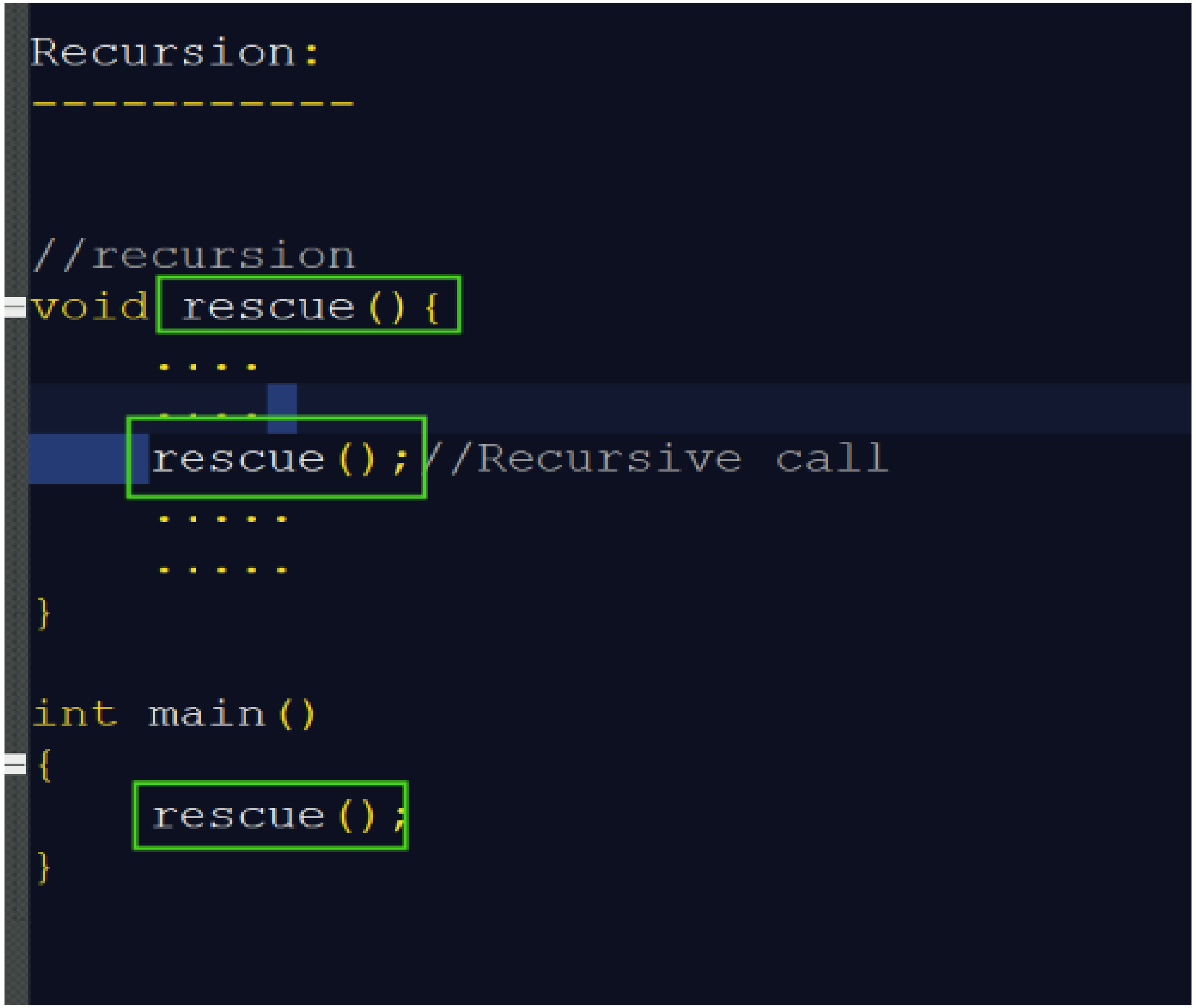
Recursion



How does Recursion works?



```
void recurse()
{
    ... .. ...
    recurse();
    ... .. ...
}

int main()
{
    ... .. ...
    recurse();
    ... .. ...
}
```



```
Recursion:
-----------


//recursion
void rescue(){
    ....
    ....
    rescue();//Recursive call
    .....
    .....
}

int main()
{
    rescue();
}
```

Recursion

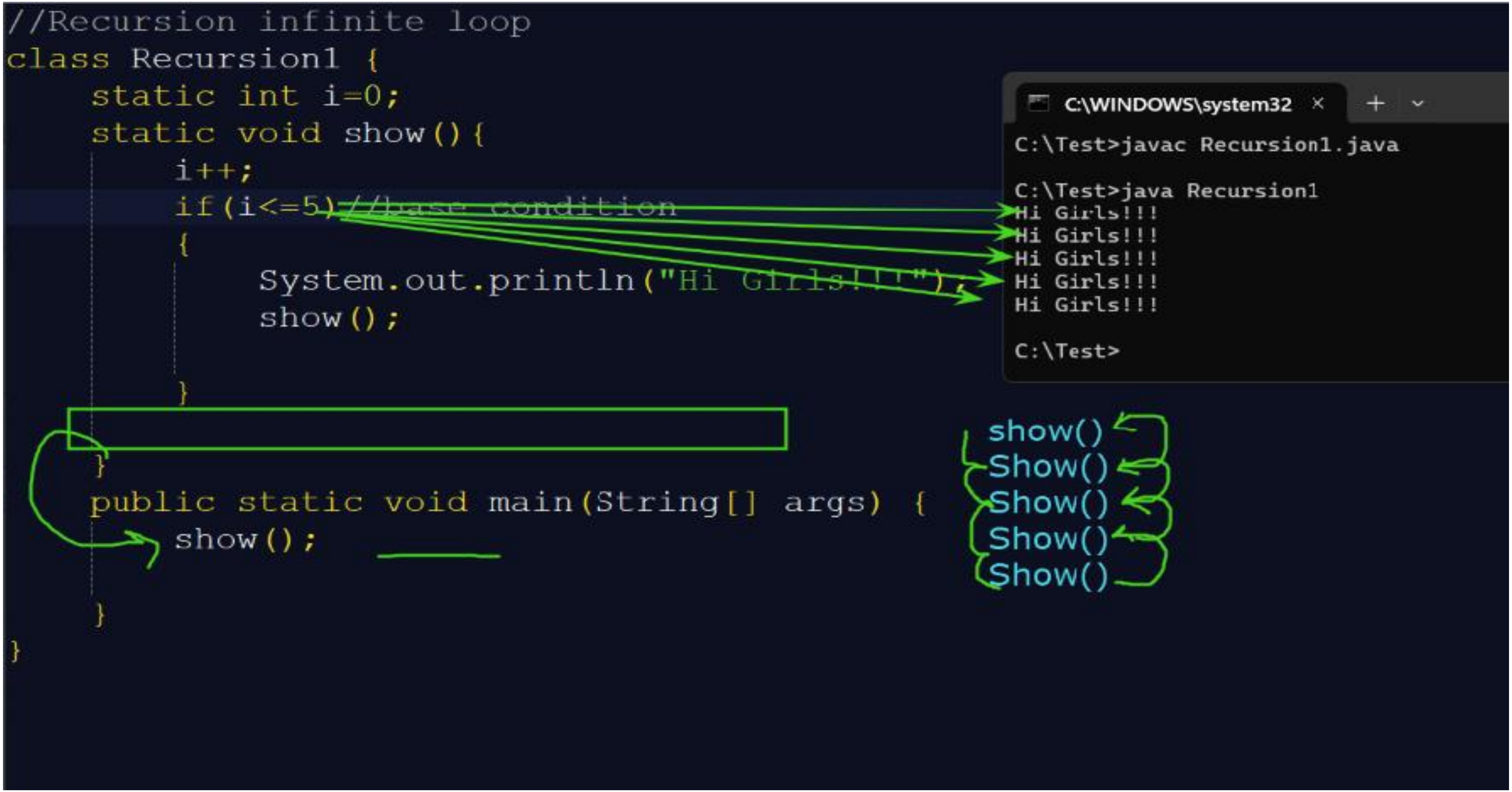
- Any function which calls itself directly or indirectly is called Recursion and the corresponding function is called as recursive function.
- A recursive method solves a problem by calling a copy of itself to work on a smaller problem.
- It is important to ensure that the recursion terminates.
- Each time the function call itself with a slightly simple version of the original problem.
- Using recursion, certain problems can be solved quite easily.
- E.g: Tower of Hanoi (TOH), Tree traversals, DFS of Graph etc.,

What is base condition in recursion?

- In the recursive program, the solution to the base case is provided and the solution of the bigger problem is expressed in terms of smaller problems.

```
int fact(int n)

{

    if (n < = 1) // base case

        return 1;

    else

        return n*fact(n-1);

}
```

- In the above example, base case for n < = 1 is defined and larger value of number can be solved by converting to smaller one till base case is reached.



```
//Recursion infinite loop
class Recursion1 {
    static int i=0;
    static void show(){
        i++;
        if(i<=5)//base condition
        {
            System.out.println("Hi Girls!!!");
            show();

        }

    }
    public static void main(String[] args) {
        show();

    }
}
```

```
C:\Test>javac Recursion1.java

C:\Test>java Recursion1
Hi Girls!!!
Hi Girls!!!
Hi Girls!!!
Hi Girls!!!
Hi Girls!!!

C:\Test>
```