Introduction to Programming

Session Agenda

- Understanding the term program
- Programs in computers context
- Generations of Programming languages
- Programming paradisms
- Translators
- Applications of Programming
- Difference between human and computer thinking
- Algorithm, psedo code, source code
- Practice Session
- Components of a Computer Program
- Q&A

Introduction
to
Programming

What is called a Program, in general?

Hi, what is your program today?

.. what are you going to do for today?

some task

or

activity..?

Program - in the computer world?

sequence of instructions that we give for the computers
to do something ..

Programming?

The process of converting the ideas into instructions for the computer.

Note: the computer must understand and execute the instructions to perform specific tasks.

Programming language..

- a means to communicate with the computer

Low level languages:

- Machine level langauage (binary) and assembly (mnemonics) are examples of low-level languages.
- Low-level languages are closer to the machine than to human-readable languages.
- high level languages: that acts as a bridge between hunams and computers - source code

Some of the major programming paradigms:

Procedural Programming:

- organizes code into procedures or routines.
- It focuses on the concept of procedure calls.
- C, Pascal, Python

Imperative Programming:

- focuses on how to perform tasks by providing a sequence of commands
- emphasises control flow
- C, Java, Python support imperative style of programming

.. programming paradigms

Object Oriented Programming:

- based on the concept of "objects,"
- promotes code reuse through inheritance and polymorphism.
- Java, Python, C++

Declarative Programming:

- you describe *what* you want to achieve without explicitly listing the steps to achieve it
- The focus is on the desired outcome rather than the control flow.
- HTML, SQL

Functional Programming:

- It emphasizes first-class functions and higher-order functions.
- avoids changing state and mutable data
- Java, Python, Javascript supports functional style

Types of translators

Compiler Vs Interpreter

compiler: translates source code to machine code at once

Interpreter: translates and executes step by step

Real-World Applications of Programming

- **Web Development:** Creating websites and web applications.
- **Mobile Apps:** Developing apps for smartphones and tablets.
- **Data Science:** Using programming for data analysis, machine learning, and visualization.
- **Games:** Designing and programming video games.

How to become a good programmer..

how to make programming simple

- tuning our thinking with computer way..

The difference between human thinking and the ways computers solve problems.

Ex: counting the students in the class

Problem statement:

Find if the number 6 is even or odd

Your answer?

How do you get it done through computer?

Problem solving strategy using the computers..

1. understand the problem

2. plan the steps that can result in the solution to the given problem (algorithm)

3. implement the plan as a computer program in a programming language

4. run the program to get the answer

Sequence of steps: (Algorithm)

1. calculate the reminder by dividing 6 with 2
2. check if reminder is zero
3. if reminder is zero print the number 6 is even
4. if reminder is not zero print the number odd

Psedu code

```
rem = 6%2
if rem == 0 :
        print('6 is even')
 else :
        print('6 is odd')
```

Characteristics of an algorithm:

- Clear problem statement, input and output
- Clear and unambiguous steps which cannot be further divided into sub tasks
- Order of steps matter
- Should complete in finite amount time
- Should produce a result

Let's put into practice ...

write an algorithm and psedo code for the following problems

- Given the age of a person, WAP that prints the year in which he will become 100 years old.

- Find the last digit of a given number

Important aspect of Problem Solving skill:

Ability to convert an idea into a sequence of algorithmic steps

Essentially breaking down a bigger problem into actionable steps that can be given to the computer to arrive at a solution

Let's do some practice ...

write algorithm and psedo code for the following problems

1. Check if the given number is positive or negative or zero
2. Find bigger of given two values
3. Find the biggest of given three values

Questions?

What does a computer program consist of?

Values & Sequence of instructions

~data structures & algorithms

Data Structures

- Data : Values   Structures : well organised
- causes the operations that we perform on the data as efficient

Types of Data Structures

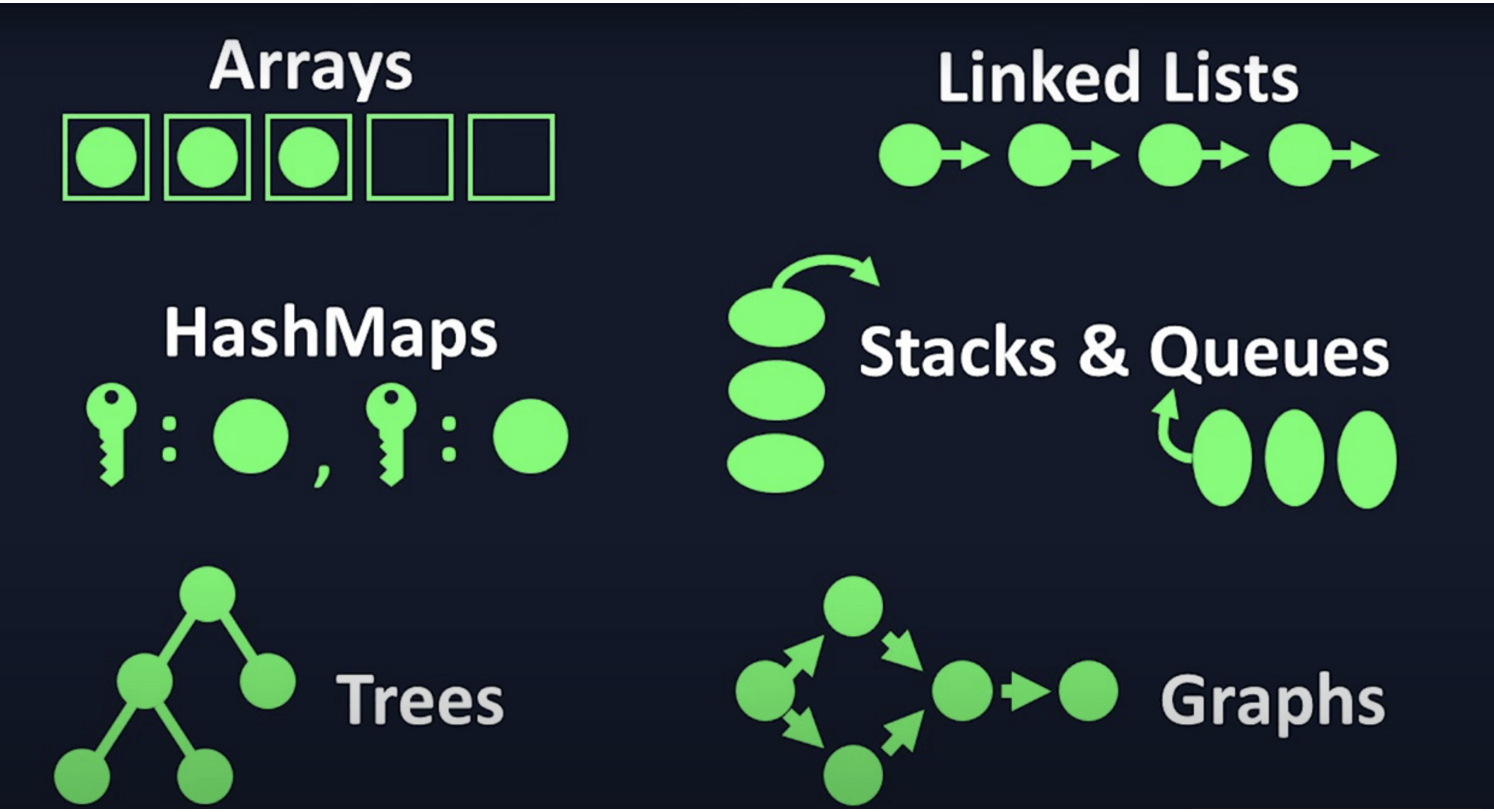
Linear data structures

- are data structures where elements are arranged in a sequential order (arrangemet is diff. from storing)
- each element is connected to its previous and next element
- Ex: Arrays, Linked Lists, Stacks and Queues

Non-Linear Data Structures

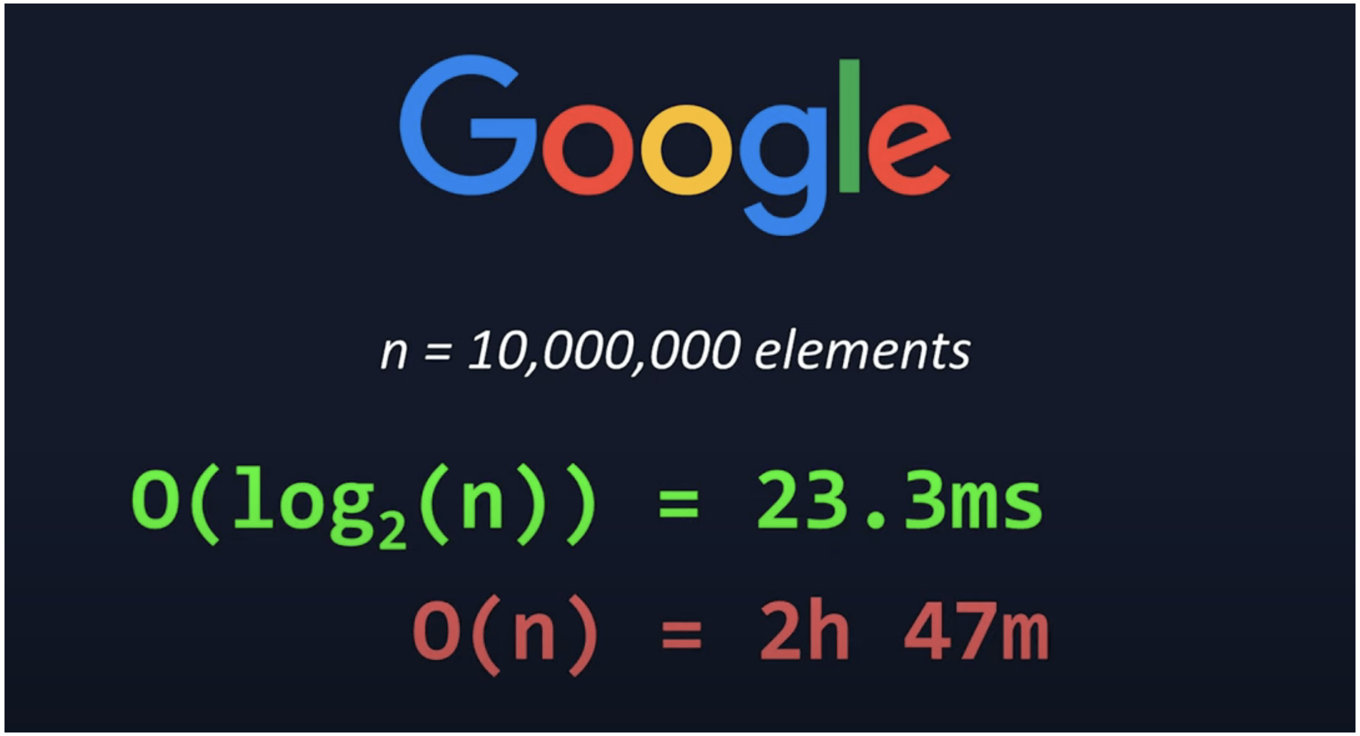
- Non-linear data structures are data structures where elements are not arranged in a sequential manner.
- Elements may have multiple relationships and form a hierarchy or a network.
- Ex: Tree, Heap, Graph

Popular Data Structures



Importance of Data Structures and algorithms in problem solving

Using appropriate data structures and efficient algorithms makes a program more performant.


Google
$n = 10{,}000{,}000$ elements
$O(\log_2(n)) = 23.3\text{ms}$
$O(n) = 2\text{h } 47\text{m}$

Programming Constructs

the fundamental building blocks used to
create programs

Basic Components of a Program

- **Variables:** Storage locations for data values (e.g., numbers, text).
- **Data Types:** Types of data that can be stored in variables (e.g., integers, floats, strings).
- **Operators:** Symbols for performing operations on variables (e.g., + for addition, == for comparison).
- **Expressions:** Combinations of variables and operators that evaluate to a value.

Input/Output (I/O)

- **Input:** How to get data from the user.
- **Output:** How to display results to the user.

Control Structures

- **Conditionals (if-else):** Allow decision-making in a program based on conditions.
- **Loops (for, while):** Repeat a set of instructions multiple times.

Function

Definition: Reusable block of code that perform specific task.

Thank you !