UCEST105

ALGORITHMIC THINKING WITH PYTHON

MODULE - 1

Not a technical definition !!!

Programming basically is getting your computer to do the stuff

Programming language

Programming involves creating a sequence of instructions, or *code*, that a computer can execute to tackle computational problems.

The language used to write this code is known as a *programming language*.

Example:

C, C++, Java

History of Python

- ❖Conceived in the late 1980s as a successor to the ABC language and its implementation was started in December 1989
- ❖Developed by **Guido van Rossum**, first released in 1991.

Monty Python, the English comedy troupe - inspired Python’s name.

Python Features

- General purpose programming language which can be used for both scientific and non scientific programming.
- Simple high level language with vast library
- Excellent for beginners as the language is interpreted, hence gives immediate results.
- The programs written in Python are easily readable and understandable.

Algorithm

An algorithm is a set of well-defined steps and instructions that can be translated into a form, usually known as the code or program, that is executable by a computing device.

When an algorithm is executed on a set of inputs, it will proceed through a set of well-defined states and eventually produce an output

Hungarian mathematician George Pólya in the famous book “How To Solve It”

Four phases a problem solver:

1) Understand the problem and clarify any doubts about the problem statement
2) Find the connection between the data and the unknown, to make out a plan for the solution.
3) Carry out our plan, checking the validity of each step
4) Review the solution

Phase 1: Understanding the problem

What is the unknown? What are the data?

– What is the condition? Is it possible to satisfy the condition? Is the condition sufficient to determine the unknown? Or is it insufficient? Or redundant? Or contradictory?

– Draw a figure. Introduce suitable notation.

– Separate the various parts of the condition. Can you write them down?

Phase 2: Devising a plan

- Have you seen it before? Or have you seen the *same problem in a slightly different form?*

- Do you know a related problem?

- Look at the unknown! Try to think of a familiar problem having the same or similar unknown.

- Split the problem into smaller, simpler sub-problems.

- If you cannot solve the proposed problem try to solve first some related problem. Or solve a more general problem. Or a special case of the problem. Or solve a part of the problem.

Phase 3: Carrying out the plan

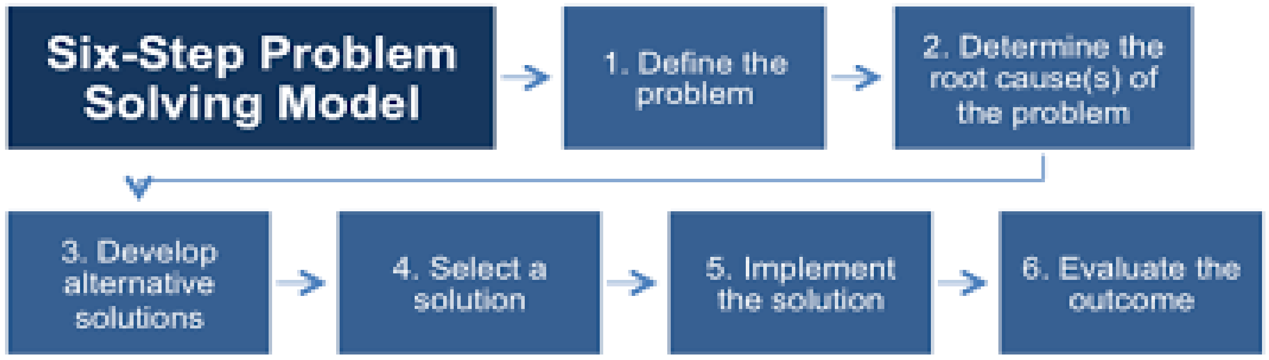
- Carrying out your plan of the solution, check each step.
- Can you see clearly that the step is correct?
- Can you prove that it is correct?

Phase 4: Looking back

- Can you check the result?
- Can you derive the result differently?
- Can you use the result, or the method, for some other problem?

Six steps of problem solving

1) Identifying the problem
2) Understanding the problem
3) Identifying the alternative solutions
4) Selecting the best solution
5) Preparing the list of instructions
6) Evaluating the solution


Six-Step Problem Solving Model
1. Define the problem
2. Determine the root cause(s) of the problem
3. Develop alternative solutions
4. Select a solution
5. Implement the solution
6. Evaluate the outcome

Algorithmic Thinking

Process of solving problems by breaking them down into a series of precise, step-by-step instructions or procedures (algorithms).

Associated with computer science and programming, but is applicable in any situation where stepwise solutions are needed.

Key Focus: Sequential steps, patterns, and structure to solve problems.

Algorithmic Thinking -contd

Example:

Problem: Making a cup of tea.

Algorithmic Approach: Break the task into a series of steps:

- Boil water.
- Place a tea bag in a cup.
- Pour boiling water into the cup.
- Remove the tea bag.
- Add sugar/milk if needed.
- Stir and serve.

Logical Thinking

Involves reasoning and making connections between ideas or concepts.

It is about evaluating situations, identifying relationships, and determining the validity of conclusions or solutions based on the evidence and principles of logic.

Key Focus: Decision-making based on reasoning.

Logical Thinking -contd

Example:

Problem: We see that the ground is wet.

Logical Approach: Deduce possible reasons why.

It could have rained.

A sprinkler might have been on.

Someone may have spilled water.

By considering available evidence, we reason that it rained because we also notice clouds in the sky and drops of water on leaves. Here, logical thinking helps us infer a conclusion from available facts.

Computational Thinking

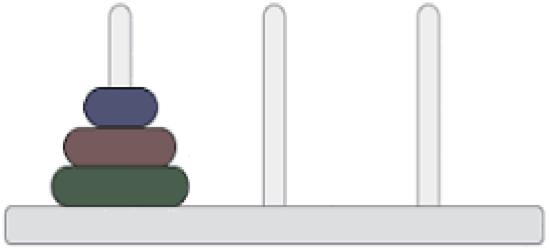
Provides the foundation for learning how to translate any real world problem into a computer program or system through the use of algorithms, languages and coding.

Algorithmic thinking is part of **computational thinking**

Tower of Hanoi can be considered both a **logical** and a **computational** problem

Tower of Hanoi

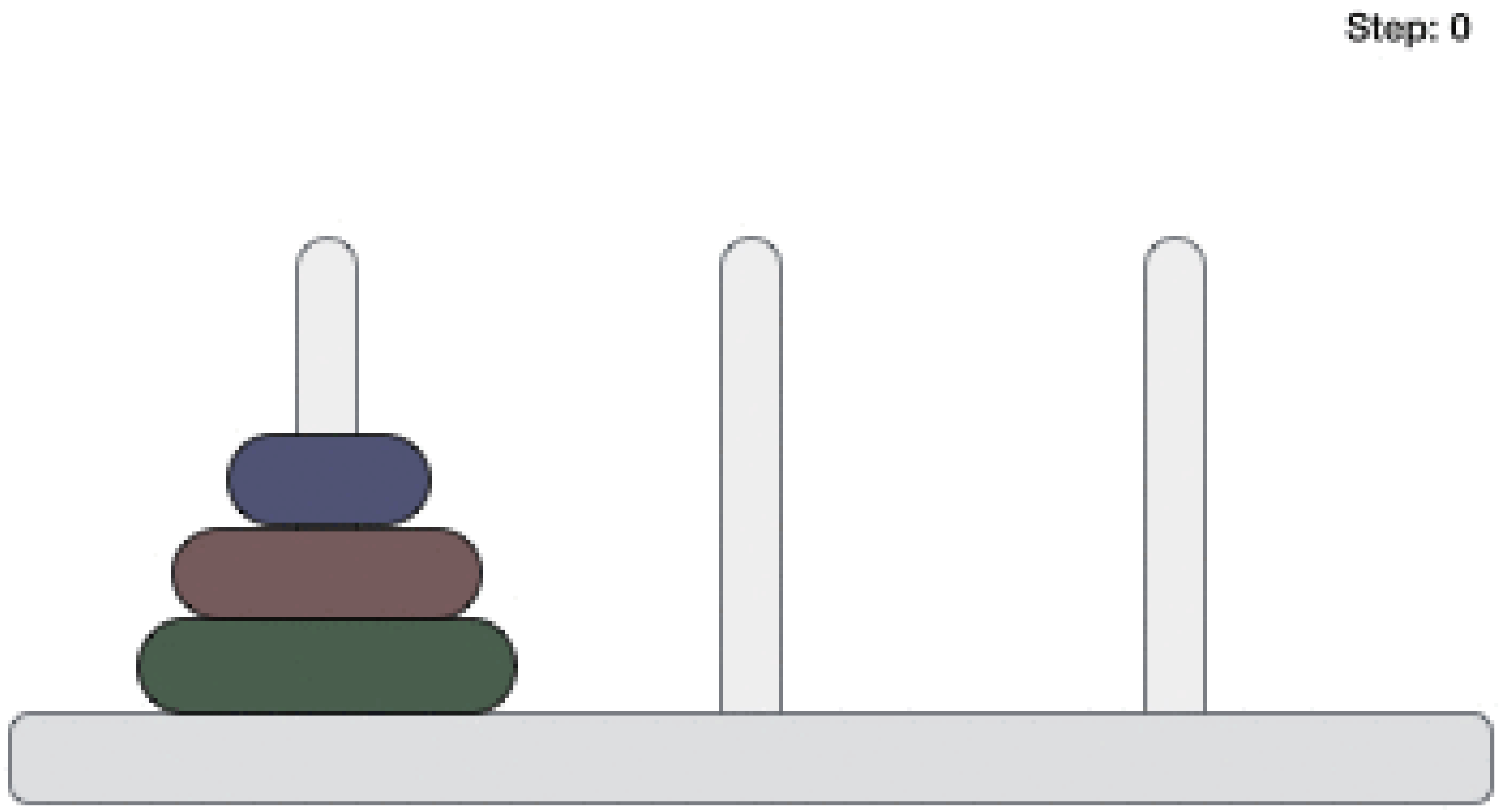
Tower of Hanoi is a mathematical puzzle where we have three rods **A**, **B**, and **C**. Initially, all the disks are stacked in decreasing value of diameter i.e., the smallest disk is placed on the top and they are on rod **A**. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:



Only one disk can be moved at a time.

No disk may be placed on top of a smaller disk.

Tower of Hanoi - Solution



Suppose that you are to ship **a wolf, a sheep, and cabbage** across the river. The boat can only hold the weight of two: you plus another item. You are the only one who can row the boat.

The wolf must not be left with the sheep alone; the wolf will eat the sheep.

Neither should the sheep be left alone with the cabbage.

How do you get them over

to the other side of the river?

Solution Using Logical Thinking

1. Take the **sheep** across the river first.
2. Go back alone to the starting side.
3. Take the **wolf** across the river.
4. Bring the **sheep** back with you to the starting side.
5. Take the **cabbage** across the river.
6. Go back alone to the starting side.
7. Finally, take the **sheep** across the river.

Module 1 - Part I

PROBLEM-SOLVING STRATEGIES: - Problem-solving strategies defined, Importance of understanding multiple problem-solving strategies,

Trial and Error,

Heuristics,

Means- Ends Analysis, and

Backtracking (Working backward)

Problem-Solving Strategies

Problem-solving strategies are essential tools that enable to effectively tackle a wide range of challenges by providing structured methods to analyze, understand, and resolve problems. Strategies include approaches

➢ Trial and Error	➢ Means-Ends Analysis
➢ Heuristics	➢ Backtracking

Problem-Solving Strategies -contd

Employing various strategies enriches

- cognitive skill development
- critical thinking
- creativity.

By integrating these strategies into everyday problem-solving, ultimately achieves more successful and innovative solutions

Importance of Understanding Multiple Problem-Solving Strategies

Multiple strategies allows for greater flexibility and adaptability.

For example, problems not solved through a trial and error method, might be solved by analytical approach like means-ends analysis.

By knowing several strategies, one can switch tactics when one method does not work.

Importance of Understanding Multiple Problem-Solving Strategies -contd

Out of the box ideas and considering multiple perspectives - can lead to more innovative and effective solutions.

Helps in recognizing patterns and similarities

between different problems - making it easier to apply previous knowledge to new situations.

Benefits of Multiple Problem-Solving Strategies

Adaptability: Different problems require different approaches. Understanding multiple strategies allows for flexibility and adaptability in problem solving.

Efficiency: Some strategies are more effective for specific types of problems. Having a repertoire of strategies can save time and resources.

Improved Outcomes: Diverse strategies offer multiple perspectives and potential solutions, increasing the likelihood of finding optimal solutions.

Skill Development: Exposure to various strategies enhances cognitive skills, critical thinking, and creativity.

Trial And Error Problem-Solving Strategy

- Attempt different solutions and learn from mistakes until a successful outcome is achieved. - Fundamental method that relies on experimentation and iteration, rather than systematic or analytical approaches.

- Initial Attempts
- Learning from Mistakes
- Refinement
- Success

Ex: Recover forgotten p/w

This approach allows for discovering the correct answer through persistence and adaptability

Heuristic Problem-Solving Strategy

A heuristic is a practical approach to problem-solving based on experience and intuition.

When driving we use a heuristic approach to find the fastest route to your destination:

1. **Rule of Thumb:** We know from experience that certain streets are typically less congested during rush hour.
2. **Current Conditions:** We use a traffic app to check current traffic conditions, looking for red or yellow indicators on major roads.
3. **Alternative Routes:** You consider side streets and shortcuts we have used before that tend to be less busy.
4. **Decision:** Based on the app and our knowledge, avoid the highway (heavy congestion) and take routes that have lighter traffic.

Heuristics approach combines your experience and real-time data

Means-Ends Analysis Problem-Solving Strategy

Involves breaking down a problem into smaller, manageable parts (means) and addressing each part to achieve the final goal (ends).

It involves identifying the current state, the desired end state, and the steps needed to bridge the gap between the two.

Means-Ends Analysis Problem-Solving Strategy - contd

- Define the Goal
- Analyze the Current State
- Identify the Differences (start, destination)
- Set Sub-Goals
- Implement the Plan
- Adjust as Needed

This method ensures that you stay on track and make steady progress toward your final destination, despite the complexity

Backtracking

- Backtracking is a core concept in computer science and is particularly useful when working with decision trees
- We make some sort of choice if that choice is wrong we 'backtrack' - make a different choice and move forward from there until we get a correct solution
- Backtracking can be used to solve puzzles like Sudoku, the eight queens puzzle, crosswords, maze

This method can be slow, especially if there are many solutions to check