SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 19CST101 – PROGRAMMING FOR PROBLEM SOLVING

I YEAR/ I SEMESTER

UNIT – I INTRODUCTION TO PROBLEM SOLVING TECHNIQUES

Topic: Algorithmic Problem Solving

Mrs.B.Sumathi
Assistant Professor
Department of Computer Science and Engineering

Algorithmic Problem Solving

Algorithmic problem solving is solving problem that require the formulation of an algorithm for the solution

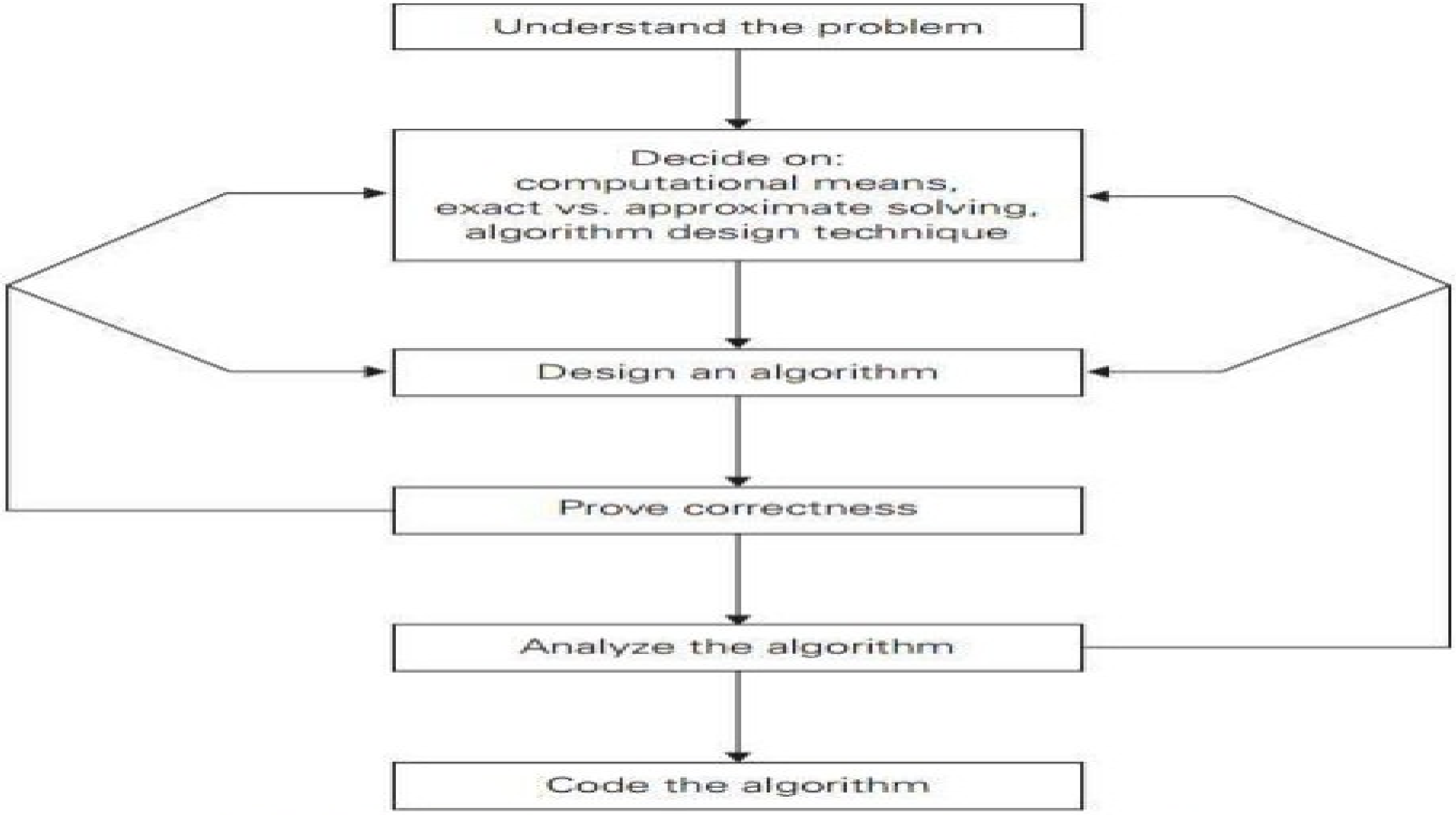


FIGURE 1.2 Algorithm design and analysis process.

Algorithmic Problem Solving

1.Understanding the Problem

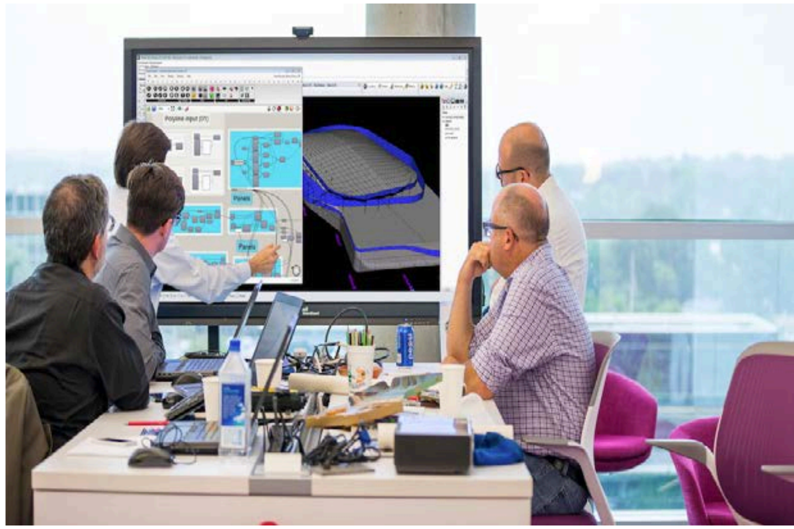
- It is the process of finding the input of the problem that the algorithm solves.
- It is very important to specify exactly the set of inputs the algorithm needs to handle.

2.Ascertaining the Capabilities of the Computational Device

- If the instructions are executed one after another, it is called sequential algorithm.
- If the instructions are executed concurrently, it is called parallel algorithm.

3.Choosing between Exact and Approximate Problem Solving

- To choose between solving the problem exactly or solving it approximately.
- Based on this, the algorithms are classified as exact *algorithm* and *approximation algorithm.*



Algorithmic Problem Solving

4.Deciding a data structure:

- Data structure plays a vital role in designing and analysis the algorithms.
- Algorithm+ Data structure=programs.

5.Algorithm Design Techniques

- Learning these techniques is of utmost importance for the following reasons.
- First, they provide guidance for designing algorithms for new problems.
- Second, algorithms are the cornerstone of computer science.

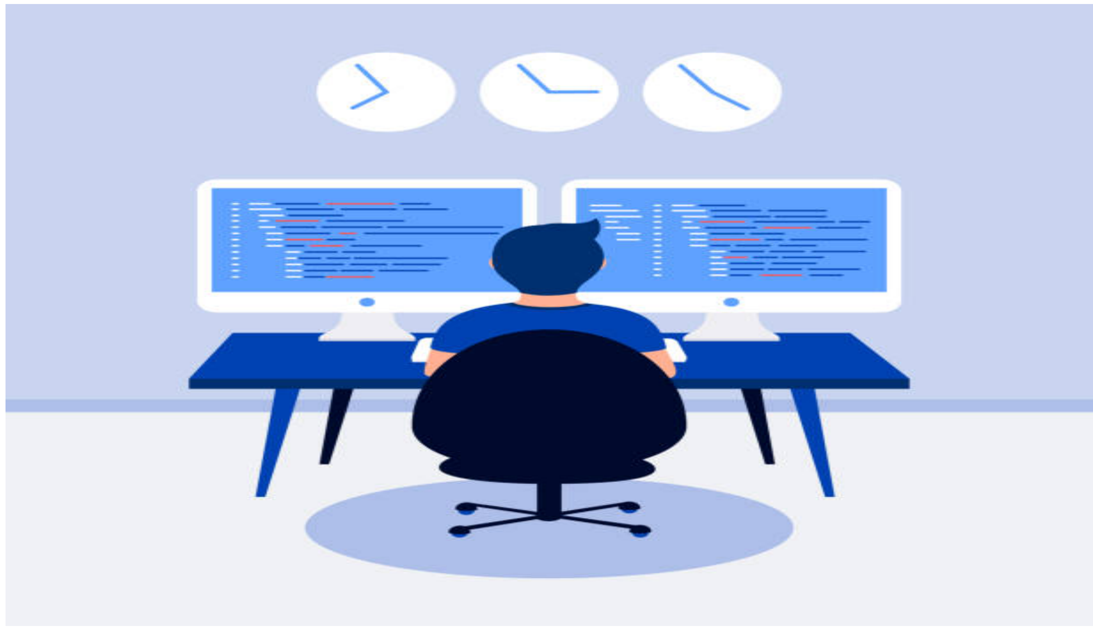
6.Methods of Specifying an Algorithm

- *Pseudocode*
- *Flowchart*
- Programming language

Algorithmic Problem Solving

7.Proving an Algorithm's Correctness

- Once an algorithm has been specified, you have to prove its ***correctness***.
- A common technique for proving correctness is to use mathematical induction because an algorithm's iterations provide a natural sequence of steps needed for such proofs.
- It cannot prove the algorithm's correctness conclusively.



8.Analysing an Algorithm

- ***Efficiency***.
    1. *Time efficiency*
    2. *Space efficiency*
- ***simplicity***.

9.Coding an Algorithm

- Most algorithms are destined to be ultimately implemented as computer programs. Programming an algorithm presents both a peril and an opportunity.

Thank you