Problems and Approach

Computational Problem

Definition: A task that can be solved using a set of well-defined steps by a computer.

There are various kinds of computational problems:

- **Mathematical**
  - Finding the factorial
  - prime numbers
  - sum of `n` integers
- **Data Processing**
  - Filter, search, and sorting a list
- **Graphing and Network**
  - Cycles in a network
  - shortest path between two points
- **Optimization**
  - Scheduling and Time table
  - cost effective containers
- **Real Life**
  - Converting file types
  - cryptography

Solving a Problem

It is important to approach problems in the right way in order to build an efficient solution in less time, and also flexible enough for future changes. Here are the 7 fundamental steps to solve a problem efficiently:

1. **Understand the Problem**

- It is important to know what the problem is exactly about, what has been given and what has to be found.
- This stage is about knowing the input and output.

2. **Analyse the Problem**
    - Divide the problem into sub-problems, and find out what operations can be used, what are the relations between data and what kind of data must be used.
3. **Design Solution**
    - Create the possible approach to the problem in the form of an algorithm, flowchart or a pseudocode, depending on what makes it easier to implement.
4. **Implement / Code**
    - Write code or implement the designed solution to bring it to life. Make sure to follow syntax and coding standards.
5. **Test / Debug**
    - Once the program has been written, test it using a variety of test cases. Make sure to test it enough so that it handles almost any situation correctly.
    - If any error has been spotted (logical or run-time), make sure to go back and check for the error and correct it. This is called debugging.
6. **Evaluate**
    - Check for time wise complexity and space wise complexity. i.e. Whether it is time wise efficient and space wise efficient even when unusual data has been entered.
7. **Documentation / Maintanace**
    - Document everything related to the problem — What it is about, input, output, designed solution, code and outputs of important test cases.
    - Update or improve the solution when requirements change.

Summary

- A computational problem is any task solvable by a computer through defined steps. Common types include mathematical, data processing, graph/network, optimization, and real-life tasks.
- Effective problem solving follows 7 steps:

  1) Understand inputs and required outputs

  2) Analyse by decomposing into subproblems and operations

  3) Design the solution using an algorithm, flowchart, or pseudocode

  4) Implement by writing clean, standard-compliant code

  5) Test with diverse cases and debug issues found

  6) Evaluate time and space efficiency, including edge cases

  7) Document thoroughly and maintain as requirements evolve.
- Choosing the right representation at the design stage speeds up coding, reduces bugs, and makes later maintenance easier.