Introduction to Problem Solving

Outline

- The problem solving aspect
- Problem solving steps
  - Problem definition phase
  - Getting started on problem
  - The use of specific examples
  - Similarities among problems
  - Working backward from solution
- General problem solving strategies

The problem solving aspect

- Regardless of the area of the study, computer science is all about solving problems with computers. The problem that we want to solve can come from any real-world problem or perhaps even from the abstract world.
- Computer cannot solve a problem by themselves. It solves a problem on basis of step by step instructions given by us.
- Computer programmers are problem solvers.
- In order to solve a problem on a computer, we must know how to represent the information describing the problem and determine the steps to transform the information from one representation into another.

The problem solving aspect

- Computers being just a machine, are not intelligent to solve a problem.
- These machines are capable of following the instructions with high computational power and automation.
- Hence, for each problem we need to follow a standard set of steps to formulate the solution. These solutions are known as "algorithms".
- Programmers formulate the solutions and convert them into a form which the computer systems can understand using programming languages. \

Solution in form of algorithm

Solution in form of computer program

Steps to solve a problem with a computer

STEP 1 – Understanding the problem

- Understanding the problem is half the solution.
  - You appearing for some exam. You are not able to understand the question
    - Can you find solution?? ⍰ **NO**
- Here we try to understand the problem to be solved in totally. Before with the next stage or step, we should be absolutely sure about the objectives of the given problem.

Steps to solve a problem with a computer contd...

STEP 2 – Analyzing the problem

- After understanding thoroughly the problem to be solved, we look at different ways of solving the problem and evaluate each of these methods.
- The idea here is to search for an appropriate solution to the problem under consideration.
- Analysing the problem helps to figure out
  - What are the inputs that our program should accept?
  - What are the outputs our program should produce?

Steps to solve a problem with a computer contd...

STEP 3 – Developing the solution

- Here, the overview of the sequence of operations that was the result of the analysis stage is expanded to form a detailed step by step solution to the problem under consideration.
- It is essential to device a solution before writing a program code for a given problem i.e. to form an **"Algorithm".**
- **Algorithm**: A set of exact steps which when followed solves the problem or accomplish the required task.

Steps to solve a problem with a computer contd...

STEP 3 – Developing the solution contd..

- Every algorithm should have the following five characteristics:
  - **Definitiveness**: Each step must be defined precisely.
  - **Effectiveness**: Its operations must be basic enough to be able to done exactly and in finite length of time.
  - **Termination:** Must terminate after a finite number of steps.
  - **Input:** Should take desired input.
  - **Output:** Should provide desired output.

Steps to solve a problem with a computer contd...

STEP 3 – Developing the solution contd..

- Example of an Algorithm- Riding a Bicycle
  - Remove the bicycle from the stand
  - Sit on seat of the bicycle
  - Start peddling
  - Use breaks whenever needed
  - Stop on reaching the destination

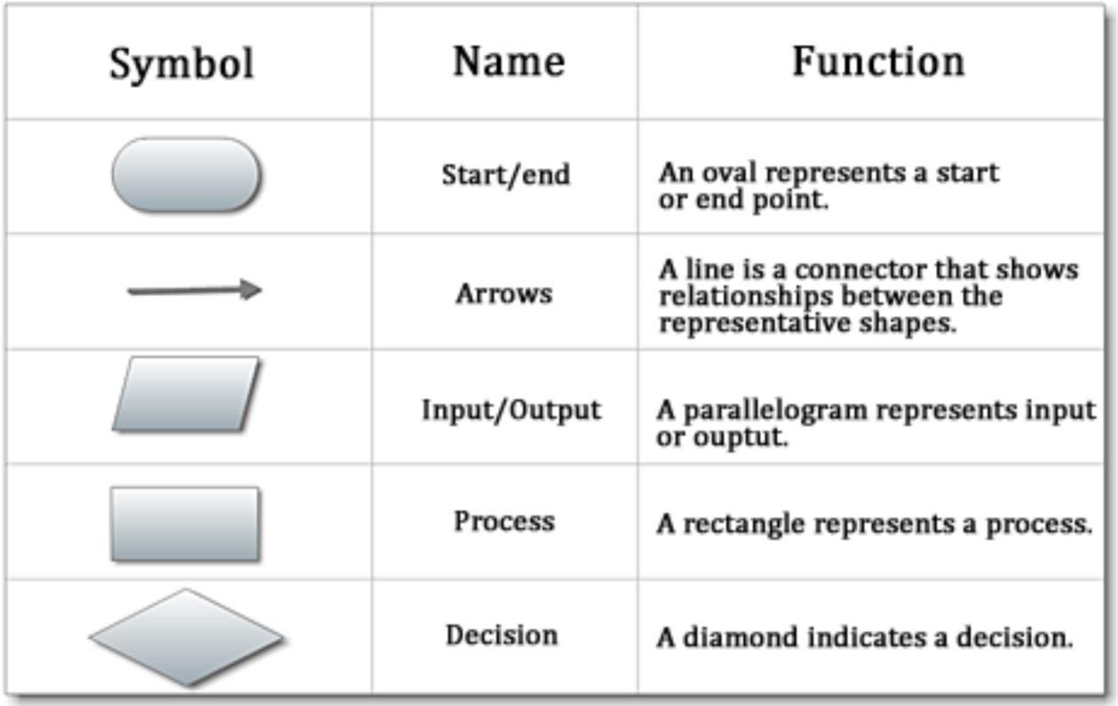
Steps to solve a problem with a computer contd...

STEP 3 – Developing the solution contd..

❖ Representation of Algorithms

a. **Flowchart** – Visual representation of Algorithms

- A flowchart of diagram made up of boxes, diamonds and other shapes connected by arrows
- Each shape represents a step of solution process and arrow represents the order or link among the steps.

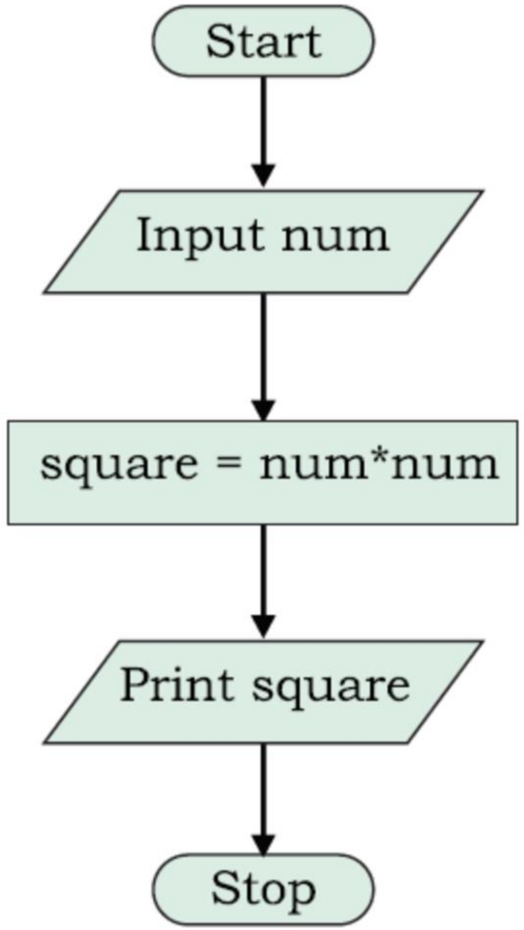


Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

Steps to solve a problem with a computer contd...

STEP 3 – Developing the solution contd..

- Representation of Algorithms
  - E.g. Flow chart of algorithm to find square of a number.



Steps to solve a problem with a computer contd...

STEP 3 – Developing the solution contd..

- Representation of Algorithms

   a. **Pseudocode** – It is considered as a non formal language that helps programmer to write algorithm. It is a detailed description of instructions that a computer must follow in a particular order.

   - It is intended for human reading and cannot be executed directly by computer.
   - No specific standard for writing pseudocode exists.
   - Keywords used in pseudocode are – INPUT, COMPUTE, PRINT, INCREMENT, DECREMENT, IF/ELSE, WHILE, TRUE/FALSE

Steps to solve a problem with a computer contd...

STEP 3 – Developing the solution contd..

- Representation of Algorithms

  a. **Pseudocode E.g. –** Write an algorithm to calculate area and perimeter of rectangle.

  INPUT Length

  INPUT Breadth

  COMPUTE Area=Length*Breadth

  PRINT Area

  COMPUTE Perimeter=2*(Length + Breadth)

  PRINT Perimeter

Steps to solve a problem with a computer contd...

STEP 4 – Coding and Implementation

- The last stage of problem-solving is the conversion of the detailed sequence of operations into a language that the computer can understand.
- Here, each step is converted to its equivalent instruction or instructions in the computer language that has been chosen for the implantation.
- The vehicle for the computer solution to a problem is a set of explicit and unambiguous instructions expressed in a programming language.
- This set of instruction is called a program with **problem solving through programming in C**.

Steps to solve a problem with a computer contd...

STEP 4 – Coding and Implementation

- A program may also be thought of as an algorithm expressed in a programming language.
- An algorithm, therefore, corresponds to a solution to a problem that is independent of any programming language.
- To obtain the computer solution to a problem once we have the program we usually have to supply the program with input or data.
- The program then takes this input and manipulates it according to its instructions.
- Eventually produces an output which represents the computer solution to the problem.

Problem solving steps

STEP 1 – Problem Definition Phase

- In the problem definition phase, we must emphasize what must be done rather than how is it to be done.
- We try to extract the precisely defined set of tasks from the problem statement.

Problem solving steps

STEP 2 – Getting started on a problem

- There are many ways of solving a problem and there may be several solutions. So, it is difficult to recognize immediately which path could be more productive.
- Sometimes you do not have any idea where to begin solving a problem, even if the problem has been defined.
- Such block sometimes occurs because you are overly concerned with the details of the implementation even before you have completely understood or worked out a solution.
- The best advice is not to get concerned with the details. Those can come later when the intricacies of the problem have been understood.

Problem solving steps

STEP 3 – Use of specific examples

- To get started on a problem, we can make use of heuristics i.e the rule of thumb. This approach will allow us to start on the problem by picking a specific problem we wish to solve and try to work out the mechanism that will allow solving this particular problem.
- It is usually much easier to work out the details of a solution to a specific problem because the relationship between the mechanism and the problem is more clearly defined.
- This approach of focusing on a particular problem can give us the foothold we need for making a start on the solution to the general problem.

Problem solving steps

STEP 4 – Similarity among problems

- One way to make a start is by considering a specific example. Another approach is to bring the experience to bear on the current problems. So, it is important to see if there are any similarities between the current problem and the past problems which we have solved.
- The more experience one has the more tools and techniques one can bring to bear in tackling the given problem. But sometimes, it blocks us from discovering a desirable or better solution to the problem.
- A skill that is important to try to develop in problem-solving is the ability to view a problem from a variety of angles.
- One must be able to metaphorically turn a problem upside down, inside out, sideways, backwards, forwards and so on.

Problem solving steps

STEP 5 – Working Backwards from the Solution

- In some cases, we can assume that we already have the solution to the problem and then try to work backwards to the starting point.
- Even a guess at the solution to the problem may be enough to give us a foothold to start on the problem.
- We can systematize the investigations and avoid duplicate efforts by writing down the various steps taken and explorations made.
- Another practice that helps to develop the problem-solving skills, once we have solved a problem, to consciously reflect back on the way we went about discovering the solution.

General Problem Solving Strategies

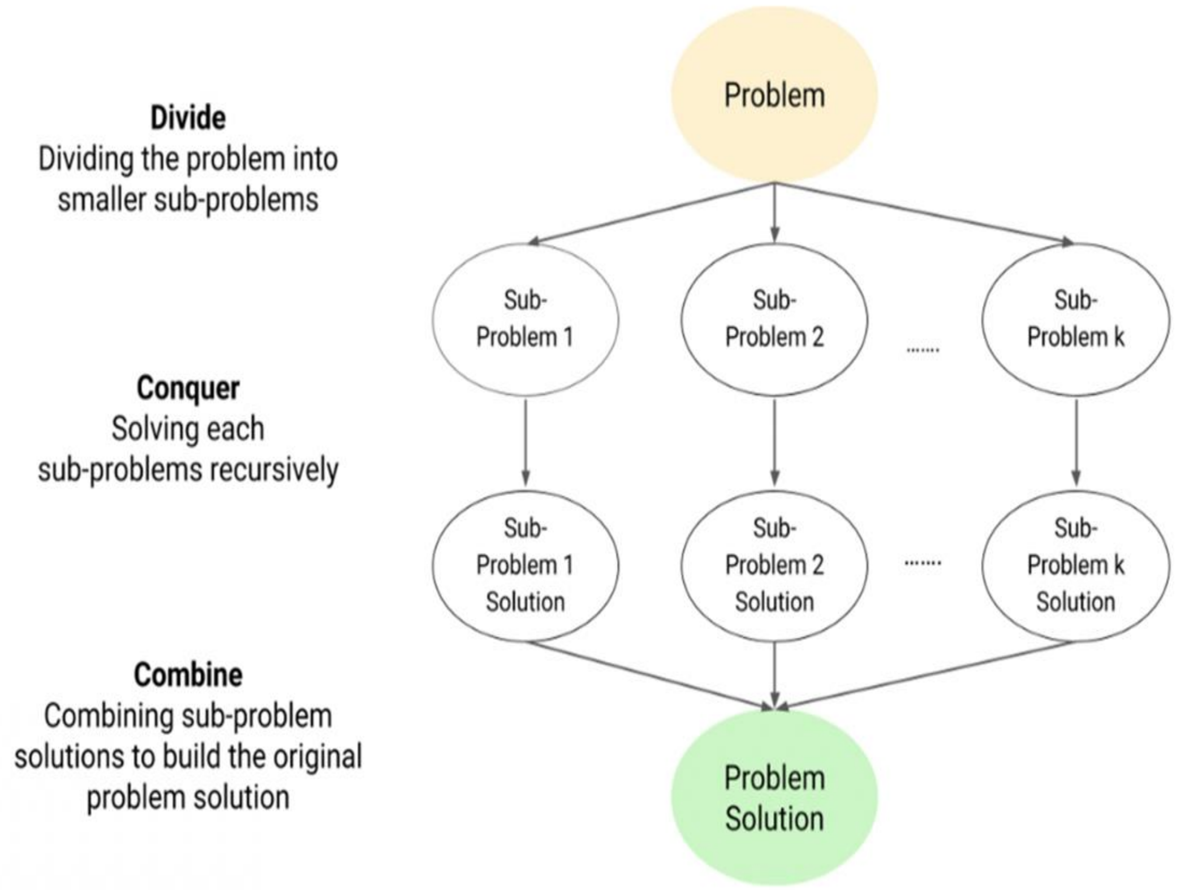
- There are a number of general and powerful computational strategies that are repeatedly used in various guises in computer science.

- Often it is possible to phrase a problem in terms of one of these strategies and achieve considerable gains in computational efficiency.

General Problem Solving Strategies

1. Divide and Conquer

- The most widely known and used strategy, where the basic idea is to break down the original problem into two or more sub-problems, which is presumably easier or more efficient to solve.
- The Splitting can be carried on further so that eventually we have many sub-problems, so small that further splitting is no necessary to solve them.



General Problem Solving Strategies

2. Binary Doubling

- This is the reverse of the divide and conquers strategy i.e build-up the solution for a larger problem from solutions and smaller sub-problems.

General Problem Solving Strategies

3. Dynamic Programming

- Another general strategy for problem-solving which is useful when we can build-up the solution as a sequence of the intermediate steps.
- The idea here is that a good or optimal solution to a problem can be built-up from good or optimal solutions of the sub-problems.
- The problem should be able to be divided into smaller overlapping sub-problem.
- An optimum solution can be achieved by using an optimum solution of smaller sub-problems.
- Dynamic algorithms use Memoization i.e. remembering the result of overlapping sub-problems.

General Problem Solving Strategies

4. General Search, Backtracking and Branch-and-Bound

- All of these are variants of the basic dynamic programming strategy but are equally important.
- Backtracking is used to find all possible solutions available to a problem. When it realises that it has made a bad choice, it undoes the last choice by backing it up. It searches the state space tree until it has found a solution for the problem.
- Branch-and-Bound is used to solve optimisation problems. When it realises that it already has a better optimal solution that the pre-solution leads to, it abandons that pre-solution. It completely searches the state space tree to get optimal solution.

References Link

- https://soniajm.medium.com/problem-solving-through-programming-in-c-5ca745b1be44
- https://learnprogramo.com/problem-solving-through-programming-in-c-1/

THANK YOU

hitbullseye