Lecture_1_Introduction

28 August 2024 06:45

Unit 1
Introduction to computer system and binary number systems – addition, subtraction (2's complement), multiplication, left shifting and right shifting.

Unit 2
Introduction to Python: Python variables, Python basic Operators, Understanding python blocks. Python Data Types, Declaring and using Numeric data types: int, float etc. Python Program Flow Control Conditional blocks: if, else and else if, Simple for loops in python, for loop using ranges, string, list and dictionaries. Use of while loops in python, Loop manipulation using pass, continue, break and else. Programming using Python conditional and loop blocks.

Unit 3
Python Complex data types: Using string data type and string operations, Defining list and list slicing, Use of Tuple data type. String, List and Dictionary.

Unit 4
Building blocks of python programs: string manipulation methods, List manipulation, Dictionary manipulation, Programming using string, list and dictionary in-built functions. Python Functions, Organizing python codes using functions, Introduction to classes.

Unit 5
Python File Operations: Reading files, Writing files in python, Case study: development of mini projects using libraries like matplotlib, numpy, etc.

Unit 1

Introduction to computer system and binary number systems - addition, subtraction (2's complement), multiplication, left shifting and right shifting.

What is computer science?

Is it only programming of computers?

Fundamentally computer science is about

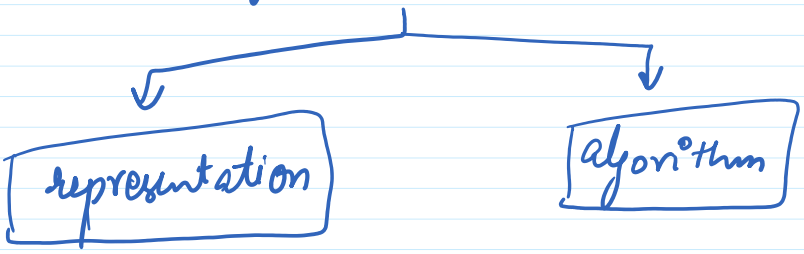
"Computational Problem Solving"

↳ Solving problems by the use of computers.

What is computation?

To solve a problem computationally

two things are needed

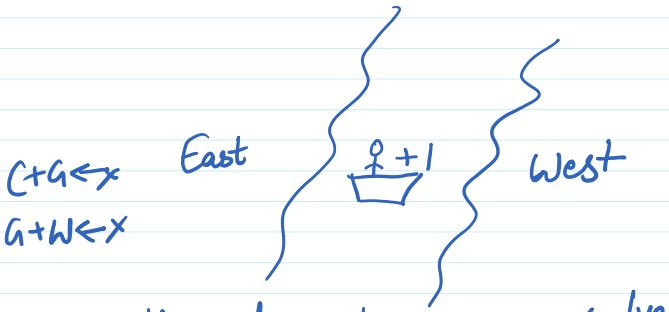


that captures all the relevant aspects of the problem.

that solves the problem by the use of the representation.

"Man, Cabbage, Goat, Wolf Problem"

[M, C, G, W]



How does the man solve his problem?

A simple algorithmic approach is solving this problem by simply trying all possible combinations Called as "brute force approach".

but is it feasible in real world?

How to represent this problem?

Start state of problem can be represented as,

man, cabbage, goat, wolf

[E, E, E, E]

East side

Let's say man took goat with him,

[W, E, W, E]

Let's say man took goat with him,

[W, E, W, E]

Solution :-

Starting State Goal State

$[\epsilon, \epsilon, \epsilon, \epsilon] \longrightarrow [w, w, w, w]$

algorithm v/s representation

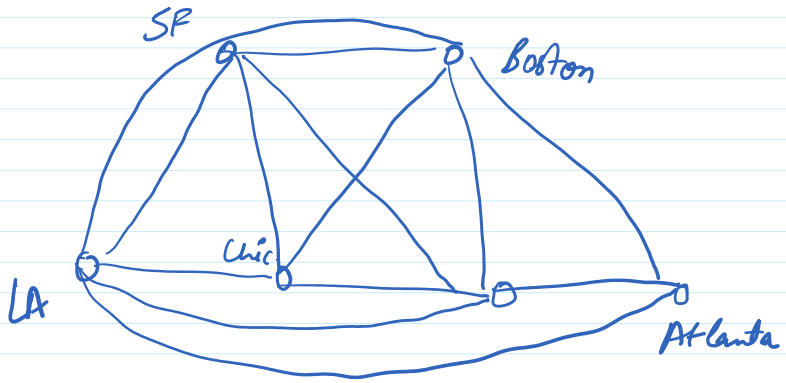
Limits of computational problem solving :-

Can a solution to the problem be found in a reasonable amount of time?

If not, is it of real-world practical use?

Different problems :-

① Travelling Salesman Problem



6 cities

② Chess :-

Any algorithm that correctly solves a given problem must solve the problem in a reasonable amount of time, otherwise it is of limited practical use.

Why Algorithms can be solved using computers?

↙

set of instructions

Because computers can execute instructions very quickly and reliably without error, algorithms and computers are perfect match.

Computer Hardware :-

Physical parts of a computer system.
All important components of the central processing unit (CPU) and main memory.

Also includes peripheral components such as a keyboard, monitor, mouse and printer.