Bangladesh Govt. & UGC Approved
UNIVERSITY OF GLOBAL VILLAGE (UGV), BARISHAL.
THE FIRST SKILL BASED HI-TECH UNIVERSITY IN BANGLADESH

Theory of Computation

Md. Mahadi Hasan Shaon
Lecturer, UGV

Course Logistics

Marking Distribution

Quiz 10%

Attendance 10%

Assignment & performance 10%

Lab Exam 20%

Total 50%

Final Grade/ Grand Total	
Midterm:	20%
Final Term:	30%
Grand Total	50%

Theory of Computation

- The ***Theory of Computation*** is the branch of computer science that deals with how efficiently problems can be solved on a model of computation, using an algorithm.
- The field is divided into three major branches:
- Automata theory and language
- Computability theory
- Complexity theory

Complexity theory

- The main question asked in this area is “What makes some problems computationally *hard* and other problems *easy*?”
- A problem is called “easy”, if it is efficiently solvable.

Examples of “easy” problems are (i) sorting a sequence of, say, 1,000,000 numbers, (ii) searching for a name in a telephone directory.

- A problem is called “hard”, if it cannot be solved efficiently, or if we don’t know whether it can be solved efficiently.

Examples of “hard” problems are (i) factoring a 300-digit integer into its prime factors.

Central Question in *Complexity Theory*: Classify problems according to their degree of “difficulty”. Give a proof that problems that seem to be “hard” are really “hard”.

Computability Theory

- Computability theory In the 1930's, G¨odel, Turing, and Church discovered that some of the fundamental mathematical problems cannot be solved by a "computer".
- To attack such a problem, we need formal definitions of the notions of *computer*, *algorithm*, and *computation*.
- The *theoretical model*s that were proposed in order to understand *solvable* and *unsolvable* problems led to the development of real computers.

Central Question in *Computability Theory*: Classify problems as being solvable or unsolvable.

Automata theory

- Automata Theory deals with definitions and properties of different types of "*computation models*". Examples of such models :
- *Finite Automata* :These are used in text processing, compilers, and hardware design.
- *Context-Free Grammars:* These are used to define programming languages and in Artificial Intelligence.
- *Turing Machines:* These form a simple abstract model of a "real" computer, such as your PC at home.

Central Question in *Automata Theory*: Do these models have the same power, or can one model solve more problems than the other?.

Theory of Computation

Purpose and motivation :

- What are the mathematical properties of computer hardware and software ?
- What is a *computation* and what is an *algorithm*? Can we give mathematical definitions of these notions?
- What are the *limitations* of computers? Can "everything" be computed?

Purpose of the TOC: Develop formal mathematical models of computation that reflect real-world computers.

Theory of Computation

Purpose and motivation :

- What are the mathematical properties of computer hardware and software ?
- What is a *computation* and what is an *algorithm*? Can we give mathematical definitions of these notions?
- What are the *limitations* of computers? Can "everything" be computed?

Purpose of the TOC: Develop formal mathematical models of computation that reflect real-world computers.