CMSC 222: Discrete Mathematics

Prof S
Spring 2019
What is Discrete Mathematics?

Discrete mathematics is the part of mathematics devoted to the study of discrete (as opposed to continuous) objects.

- Calculus deals with continuous objects and is not part of discrete mathematics.

Examples:
- Integers
- steps taken to complete a computer program
- distinct paths to travel from point A to point B on a map along a road network
- ways to pick a winning set of numbers in a lottery.
What is Discrete Mathematics?

A course in discrete mathematics provides the mathematical background needed for all subsequent courses in computer science!
Kinds of Problems Solved Using Discrete Mathematics

- How many ways can a password be chosen following specific rules?
- How many valid Internet addresses are there?
- What is the probability of winning a particular lottery?
- Is there a link between two computers in a network?
- How can I identify spam email messages?
- How can I encrypt a message so that no unintended recipient can read it?
  - Many public key cryptosystems rely on algebraic number theory
- How can we build a circuit that adds two integers?
Kinds of Problems Solved Using Discrete Mathematics

- What is the shortest path between two cities using a transportation system?
- Find the shortest tour that visits each of a group of cities only once and then ends in the starting city.
- How can we represent English sentences so that a computer can reason with them?
- How can we prove that there are infinitely many prime numbers?
- How can a list of integers be sorted so that the integers are in increasing order?
- How many steps are required to do such a sorting?
- How can it be proved that a sorting algorithm always correctly sorts a list?
Goals of This Course

- **Mathematical Reasoning**: Ability to read, understand, and construct mathematical arguments and proofs.
- **Combinatorial Analysis**: Techniques for counting objects of different kinds.
- **Discrete Structures**: Abstract mathematical structures that represent objects and the relationships between them. Examples are sets, permutations, relations, graphs, trees, and finite state machines.
Goals of This Course

Mathematical Reasoning: Ability to read, understand, and construct rigorous mathematical arguments and proofs.

- This skill is essential if you are to be a competent computer scientist!
Proofs

Proofs are NOT just a list of random true statements! And regardless of what you may have been taught, it’s not a list of equalities with question marks over them.

A proper proof tells a story. It starts at the start (hypotheses) and proceeds in a logical and clear path to the finish (result). At no point should the reader have to question ANY part of the story!
In This Course

Rule #1: IF IT HAS NO WORDS, IT’S NOT A PROOF!

A couple of equations or some scrawled inequalities does not constitute a proof.

At times this is fine. But if the instructions are to provide a rigorous proof (as they often will be), then know that an exposition is required!
Secondary Goal of this course

Algorithmic Thinking: An algorithm is a sequence of steps that can be followed to solve any instance of a particular problem. Algorithmic thinking involves

- Specifying algorithms
- Analyzing the memory and time required by an execution of the algorithm
- Verifying that the algorithm will produce the correct answer.
Discrete Mathematics is Gateway Course

Topics in this course will be fundamental in many CS courses that you will take in the future:

Most Important (Once More For Emphasis)!

- It is imperative that you learn how to read and write rigorous arguments (i.e., proofs)
  - No matter how good your work may be, it will require justification!
    - Otherwise how do you (or others) know your method works!