Queen's University
School of Computing

CISC 203: Discrete Mathematics for Computing II
Lecture 0: Introduction
Winter 2019

1 What is Discrete Mathematics?

Discrete mathematics is a subfield of mathematics that investigates, among other things,

- logical statements;
- sets and elements;
- relations and functions;
- the integers;
- counting;
- trees; and
- graphs.

When we consider some of the properties each of these topics possesses—logical statements are either true or false, sets contain individual elements, relations and functions associate sets with one another, the integers (and sets) can be counted, both trees and graphs have distinct vertices and edges—we see that one common property ties each topic together: the property of **discreteness**, or discontinuity. The word "discrete" comes from the Latin *discretus*, meaning "separate".

In discrete mathematics, we don't consider continuous topics like curves, calculus, or the real numbers. This fact makes discrete mathematics particularly agreeable with computer technology, since the fundamental unit of storage in a computer—a bit—is itself discrete; it is either `0` or `1`. Indeed, computers run into all kinds of problems, like precision issues, when they try to handle continuous mathematics. With discrete mathematics, computers have no such issues.

In this course, we will continue the study of discrete mathematics that started in CISC 102. We will begin with a brief review before investigating various **proof techniques** common in discrete mathematics. We will then proceed to **combinatorics**, or the study of counting. We take a brief detour into probability theory with a discussion on **discrete probability**. We focus on a number of mathematical structures that are easily representable in computer memory, including **trees** and **graphs**, and we consider various important properties of these structures. Finally, we explore **Boolean algebra**, skirting the border between mathematics, computer science, and computer engineering.

2 What is Discreet Mathematics?

Discreet mathematics is the kind of mathematics you don't tell your family and friends about.

3 Motivating Examples

In an effort to motivate our study of discrete mathematics, let's see a couple of examples of mathematical ideas appearing in more applied areas of computer science.

3.1 Facebook Graph API

The rise of social networking in the 21st century has made it significantly easier for people to stay at home and pretend to make meaningful connections. But how do we model such connections—not just between people, but between those people and things they like, or companies they work for, or places they visit?

We could use something like a "friend adjacency matrix", where each person on Earth is represented as a row/column and the state of "x being friends with y" is denoted by a 1 entry in row x and column y. However, it isn't exactly nice to deal with a $7\,500\,000\,000 \times 7\,500\,000\,000$ matrix.

We could instead work under the assumption that not every person on Earth can be friends with every other person on Earth, thus saving us from having to maintain entries between one person and everyone else as we did in the adjacency matrix. Instead of representing a person's friend list as a $7\,500\,000\,000$-entry vector, we could list every person in one set and the connections between every person in another (much smaller) set. This model lends itself well to the vertex/edge structure of a graph. We could even extend this model to include organizations, companies, pages, or places a person "likes" by adding that entity to our vertex set; as an added benefit, there is no need to modify the way we handle our edge set.

Indeed, graphs constitute the underlying structure of Facebook, the world's largest social networking website. According to their documentation, Facebook's Graph API allows developers to "programmatically query data, post new stories, manage ads, upload photos, and perform a wide variety of other tasks". The Graph API uses vertices (called "nodes" in the documentation) to represent users, photos, pages, and so on, and edges denote relations between nodes; for instance, photos on a user's profile.

3.2 Bitcoin

If you've turned on a television or used the Internet in the past decade, you've likely heard about Bitcoin. Bitcoin is a digital currency that is distributed via a peer-to-peer network, rather than via a central authority like a bank. It is possibly the most well-known example of a cryptocurrency: a currency that uses cryptography to secure funds and to verify transactions. The security of Bitcoin is implemented using two technologies: public key cryptography and the blockchain.

In a public key cryptography system, two users share information by encrypting and decrypting it with "keys"; essentially, keys are pairs of very large numbers. The public can encrypt messages intended for a single user with that user's public key, but only the user can decrypt the message using their private key. Therefore, once a message is encrypted, only the sender and the intended recipient will know what it says.

The Bitcoin protocol uses public key cryptography in a slightly different way: keys are used not only to transmit information, but also to verify information. The person sending Bitcoin attaches the recipient's public key to the transaction and "signs" the transaction with their private key. The recipient can then prove both that they are the new owner of the Bitcoin (due to the recipient's public key being attached to the transaction) and that the sender was truly responsible for the transaction (by using the sender's public key against the private key used to "sign" the transaction).

Since Bitcoin is a distributed system, how can we be sure that everyone has a consistent historical record of transactions? This is the job of the second technology: the blockchain. A blockchain is like a "transaction tree"; the root of the tree, called the genesis block, stores timestamp data and instructions on how to recreate the block, and other transaction blocks are added as children of a previous transaction. As a security measure, each block also contains a hashed version of the block preceding it to verify that it is located in the correct position within the blockchain.