MATH 253 – Discrete Mathematics
Exam #1 – Study Guide

Schedule

Exam #1 will be given on **Thursday, October 1, 2020. You may take the exam online in any 80-minute period between 9:00 AM and 5:00 PM. The instructor will be avilable for questions ONLY from 9:00 AM to 10:20 AM.** The exam is 80 minutes (+ accommodations).

Administrivia

This is an exam given online via Canvas. As such, there is no effective way to limit the resources you have available during the exam. I will, though, write the exam with the idea that you have a sheet of notes that you have prepared in advance. I will even make suggestions below about what that sheet of notes should have on it. But you can add whatever you think is useful to that sheet, and you are not required to turn it in. Hold onto that sheet, though, as the final exam will cover all material in the class and you can use it then, too!s

Keep in mind that the exam will be timed in a manner that does not allow extensive referrals to external materials. It will be geared towards testing your familiarity with the presented materials in class and in the online textbook. This is another good reason to condense your notes to a single "cheat sheet" you can use quickly and easily to keep pace with the problems in the time provided.

Format of exam

The majority of the questions presented will be multiple choice. There are significant limitations in giving an online exam, and one of them is the ability to write in a mathematical notation. Thus, the need for a question format like multiple choice is necessitated. These questions will not concentrate just on rote memorization, though – they are designed to allow students to show the skills of applying what has been learned in class to problems similar to, but not exactly the same as, those in the text. They are also designed to allow students to show analysis skills in order to solve problems presented to them.

You will want a pencil and lost of blank paper available to work through problems! Multiple choice doesn't mean "do it all in your head" – you'll need to write out a few things to get to many of the answers to the questions provided! There will be no need to submit the work you do on your scratch paper, though.

A multiple choice exam can be every bit as challenging as an "on paper" exam! But there will be no penalty in giving wrong answers, so answering <u>every</u> question will be to your benefit – don't leave any questions unanswered!

I will also ask you some questions that are not multiple choice, such as to present a proof, or to analyze a a truth table and answer questions based on its contents, or to answer what a given summation will do, or to provide steps required in an induction proof. These questions, though, will be far fewer than the ones in multiple choice form.

Academic Honesty

While the exam may be open book, it is ABSOLUTELY required that the work you present on the exam is your own! No real-time collaboration with other students is permitted. Canvas can randomize the order of answers listed on multiple-choice tests, and even present questions in a different order for each student, so sharing of multiple-choice answers in A/B/C/D format between students is an utterly futile effort.

Material to be Covered

You are responsible for all material covered in class and in the text from the following sections:

Chapter 1 "Logic" – All sections

Chapter 2 "Proofs" – All sections

Chapter 3 "Induction and Recursion" – All sections except for 3.10 "Solving Linear Homogeneous Recurrence Relations"

Preparation Strategies

First and foremost, the best preparation strategy is to review the material in the text, and to drill through the provided exercises. Make that your go-to strategy! Understanding the answers there, and the methods presented to get to the answers, presents your best opportunity to do well on the exam.

Topics & Terminology

You should be familiar with and be able to answer questions about:

Chapter 1 - "Logic"

- **Propositions**
- The **logical operations** of **and**, (**conjunction**), **or** (**disjunction**), **not** (**negation**), **implies** (**if/then**), and **biconditional** (**if and only if**)
  It would be a very good idea to have the basic truth tables for these on your sheet of notes!
- The use of a **compound propositions**, and **the order of operations** in a compound proposition
- The parts of a **conditional proposition**, with the **hypothesis** (the "if" part) and the **conclusion** (the "then" part) (sometimes referred to by the names **antecedent** and **consequent**)
- Being able to take a sentence in English and translate it into the **symbolic form** of propositions
- How to write the **logical negation** of a **conditional proposition**, both in English and in symbolic form, and how writing the proposition in symbolic form, negating that, then writing the negation in English can help with writing a negation more precisely
- The general forms of the **converse**, the **inverse**, and the **contrapositive** of a conditional proposition
  This is also a good thing to have on your sheet of notes, as you may refer to this often!
- **Tautologies**, **contradictions**, and **DeMorgan's Laws**
- The **Laws of Propositional Logic** – **Associative**, **Commutative**, **Distributive**, **Identity**, **Domination**, **Double Negation**, **Complement**, **Absorption**, and **Conditional Identities**
  You won't necessarily be asked to write them explicitly, but be prepared to use them as needed!
- The idea of **logical equivalence**, and how to determine whether compound propositions are logically equivalent using truth tables
- **Predicates** (or properties) written in the form of P(x) or Q(a,b), and **domains** of predicates

- **Universal quantifiers**, and what they mean in **universally quantified statements**
- **Existential quantifiers**, and what they mean in existentially **quantified statements**
- The use of a **counterexample** to show that a universally quantified statement is false
- How to write the **logical negation** of a universally quantified statement or an existentially quantified statement by the use of **DeMorgan's Laws** for such statements
- The **bound** variable(s) and the **free** variable(s) in quantified statements
- **Nested quantifiers**, and how they can create a combined quantified statement (sometimes called an **existential universal** or a **universal existential** statement)
- The use of **logical reasoning** to show the **validity** or **invalidity** of an **argument** consisting of one or more **hypotheses** and a **conclusion** by the use of **truth tables**

Chapter 2 – "Proofs"

- **Properties of integers** and their definitions you should specifically cite in your proofs when you use them – **even integer**, **odd integer**, **rational number**, an integer **dividing** another integer (and the related definition of a **multiple** of an integer), a **prime** integer, a **non-prime** (or **composite**) integer, and **consecutive** integers. More basic definitions and rules of arithmetic may be used without specifically mentioning them.
- What a **theorem** is, and what it means to **prove** or **disprove** a theorem
- The difference between a theorem and an **axiom**, and how axioms are used to prove theorems
- Proving an **existential statement** by use of a single **example** (called an **existence proof**)
- Proving an **universal statement** by use of one of several available **proof methods**
- Disproving an universal statement by use of an **counterexample**
- Disproving an existential statement by stating its negation as a universal statement, then using one of several available proof methods
- **Proof by exhaustion** for a theorem where the domain for the variable is finite
- The use of **universal generalization** to start a proof of a universal statement, by letting ("suppose that") a variable represent an "arbitrary but specific" value in the domain
- **Direct proof** of a conditional statement, where assuming the hypothesis ("if" part) is true leads directly to the conclusion ("then" part) being true
- **Proof by contrapositive**, where a conditional statement is rephrased as its contrapositive (which is logically equivalent), and then the contrapositive is proved true by another proof method
- **Proof by contradiction**, where a statement is assumed to be false, which then leads to a violation of a known property of mathematics (such as a property of integers or an axiom)
- **Proof by cases**, where a universal statement is "split up" by considering different subsets of the domain of the variable (for example, odd integers and even integers), and proving each case separately, taking care that the union of the subsets of the domain equals the entire domain

Chapter 3 – "Induction and Recursion"

- What a **sequence** is and how it can be represented in both **set notation** and **function notation**

- What **terms** are in a sequence
- What **index values** are in a sequence that represent the order of terms in a sequence
- What the following terms means for sequences – **increasing**, **non-decreasing**, **non-increasing**, **decreasing**
- What a **geometric sequence** is with a **common ratio**
- What an **arithmetic sequence** is with a **common difference**
- The basic idea of how a **recurrence relation** can be used to define the value of a term based on the value of a previous term (or terms)
NOTE: You will NOT be required to solve any Linear Homogeneous Recurrence Relations! (In other words, Section 3.10 will NOT be on the exam)
- How a sequence of numbers can be added together to make a **summation**
- The **summation form** in which summations are written using a symbolic form, with an **index variable** that has a **lower limit** and an **upper limit**
- What the **final term** is in a summation, and how to rewrite a summation that "pulls out" the final term
- How to rewrite a summation by changing the index variable and lower and upper limits without changing the values in the summation itself
- What the **closed form** of a summation is
- How to write the summation of an geometric sequence in closed form
- How to write the summation of an arithmetic sequence in closed form
- How the **principle of induction** uses a **base case** and an **induction step** to prove a universal statement that has a domain of the positive integers (or a domain with similar properties)
- How to write an **induction proof** by showing the base case(s) are true, and that the induction step is valid for an "arbitrary but specific" value in the domain by assuming the statement is true for the value K and then showing that the statement is also true for K+1
- What **strong induction** is, where the induction step to show a statement is true for value K+1 requires more than just being true for the value K (that is, it also requires the statement is true for K-1, K-2, etc.)
- For writing a loop in code, knowing what is meant by a **pre-condition**, a **post-condition**, and a **loop invariant** (something that is true at the end of each iteration of the loop)
- The **principle of well-ordering**, that states that any non-empty subset of the whole numbers (non-negative integers) has to have an element that is the "smallest" element in the subset – we can use this principle as an axiom in proofs whenever we need to refer to a set's smallest element
- A **recursive definition** that defines a function or a set by referring to itself, like a recurrence relation does – examples include the Fibonacci numbers and the recursive definition of a factorial
- How a **recursive definition** closely resembles induction with the use of a **basis**, a **recursive rule**, and an **exclusion statement** (that says anything not derived from the basis or the recursive rule is not in the defined set)
- How a **recursive algorithm** can be used to write a function to implement a recursive definition