

Study Guide for Test II

Randall Holmes

November 5, 2023

Test II will be on Friday, November 10. You will be allowed a calculator without symbolic computation or graphing capability and a single sheet of notes.

This document reviews what is covered.

I like Levin problems for test review for exactly the reason that I do not like them as homework: most of them have answers. But you have to use them sensibly: you learn by attempting them by yourself and checking your solutions.

principle of inclusion and exclusion?: This is a counting topic not covered on the first test; it will not be covered on the second test either. I don't rule out asking about it on the final.

2.1, describing sequences: Understand what a sequence is, what a closed form definition for a sequence is, and what a recursive definition of a sequence is.

You need to know sum and product notation.

Problems in Levin section 2.1 which catch my eye, 3, 4 (note the idea of sequence of partial sums), 7, 8, 9, 10, 13, 14.

2.2, arithmetic and geometric sequences: Be able to sum arithmetic and geometric series by the methods given. Also, be aware of indexing and term counting issues with arithmetic and geometric sequences (which generalize to sequences defined in other ways!)

2.2 problems 1 through 9 and 14 for study.

2.3: Polynomial fitting. Understand the method of differences and the method of finding closed forms for sequences described in the section. You are welcome to use the method using binomial coefficients which I have described (and I will review it) but it will not be required.

2.3: every problem is a good study problem!

2.4: Be able to find closed forms for recursively defined sequences using the characteristic root method. Be aware that I may ask you to verify that a sequence of the relevant kind actually satisfies the recurrence relation (a 2.1 problem) or to prove that a sequence of the relevant kind has the appropriate closed form by math induction (a 2.5 problem) and the characteristic polynomial is then beside the point as you know the closed formula in such a problem already.

2.4 study problems: 1,2,5,6,7,10,13 (yes, you have to know this case).

2.5: Be able to write math induction proofs for stated theorems.

2.5 study problems: 3,4,5,6,10,21,22,23,25,29

number theory notes: basic concepts and axioms: I am not going to ask test questions directly about this.

number theory notes: divisibility and primes: One of the two theorems about primes might appear (prove by strong induction that every natural number greater than or equal to 2 is a prime or a finite product of primes, or prove that there are infinitely many primes).

I have already asked you to prove basic theorems about divisibility, and I won't do that on this test.

number theory notes: division algorithm: I'm not going to ask you a direct question about integer division and remainder, except possibly one where a is negative, but you will certainly need to be able to compute div and mod with your calculator to do the Euclidean algorithm and modular arithmetic questions.

number theory notes: Euclidean algorithm: Be able to compute gcds and determine $\gcd(a, b)$ as a linear combination of a and b .

You need to master my table method. I know from observation that there really isn't a substitute.

Don't overuse the spreadsheet in practice.

I suggest doing lots of numerical examples, which you can readily make up.

I may ask word problems like those in the homework.

The proof of Euclid's lemma is examinable.

modular arithmetic: Be able to build a small addition and/or multiplication table for arithmetic mod some m and be able to identify and tabulate additive and/or multiplicative inverses of remainders mod m using the table.

Be able to compute multiplicative inverses mod m using the Euclidean algorithm appropriately and solve equations of the form $ax \equiv_m b$ (or explain why they have no solutions, in cases where m is composite this can easily happen).

I'll do concrete examples of all these tasks on Monday.

proofs: You may have a choice of proofs to do.