

# Discrete Structures II - Topics for Fall 2021

---

Notes:

- “Know basic terminology” means that students should have at least a superficial understanding of the given terminology. For example, this might mean that they can identify when a given object satisfies a definition or doesn’t (at least for sufficiently simple examples). Also, they should be able to justify their answers in a reasonable way.
  - For the basic terminology in the group theory chapters, students should be able to do computations for the groups  $\mathbb{Z}_n$  and  $\mathbb{Z}$ .
- 

## Section 9.1: Graphs - General Introduction

- Know basic terminology: Directed / undirected graphs, vertices, edges, degree in undirected graphs, in-degree / out-degree in a directed graph, paths, circuits, subgraphs, spanning subgraph.
- Graph isomorphism: For two isomorphic graphs, be able to give an example of an isomorphism. If two graphs are not isomorphic, explain why an isomorphism can’t exist.

## Section 9.2: Data Structures for Graphs

- Adjacency matrix: Be able to write down the adjacency matrix for a graph. Given an adjacency matrix, draw the corresponding graph.

## Section 9.3: Connectivity

- Know what connected / strongly connected means.
- Identify the connected components in a graph.
- Be able to use the breadth-first search in Algorithm 9.3.8 to finding a path between vertices in a graph.

## Section 9.4: Traversals: Eulerian and Hamiltonian Graphs

- Use Euler’s Theorem (Theorem 9.4.7) to determine if an Eulerian path and/or circuit exists.
- Be able to find Eulerian paths and circuits in graphs when they exist.
- Be able to find a Hamiltonian path or circuit when it exists.

## Section 10.1: What Is a Tree?

- Know basic terminology: cycle, tree
- Know the relationship between the number of vertices and edges in a tree.

## Section 10.2: Spanning Trees

- Know basic terminology: Spanning tree, minimal spanning tree
- In a weighted graph, compute the weight of a spanning tree.
- Be able to use Prim’s algorithm to find a minimal spanning tree.

### Section 10.3: Rooted Trees

- Know basic terminology: rooted tree, root, parent, child, ancestors, descendants, subtrees, level of a vertex, depth

### Section 10.4: Binary Trees

- Know basic terminology: Ordered rooted trees, binary trees, left / right subtrees, leaf, internal vertices
- Be able to perform preorder, inorder, and postorder traversals of binary trees.
- Write the binary sort tree for a list of numbers. The inorder traversal yields the list of numbers in increasing order.
- Write down the expression tree for an algebraic expression.

### Section 11.1: Operations

- Know basic terminology: Unary operation, binary operation
- Know the properties of operations: Commutative, associative, identity, inverse, idempotent, left / right distributive, distributive, involution, closure.
- Be able to identify and explain if an operation has a certain property or not.
- Understand and be able to construct operation tables.

### Section 11.2: Algebraic Systems

- Know basic terminology: groups and basic notation for groups, abelian group.
- Given a set with a binary operation, determine if it's a group or not. If it is, prove it. If not, explain why not.

### Section 11.3: Some General Properties of Groups

- Know the basic properties of identity and inverses (Theorems 11.3.1-9). Be able to prove the properties using the definition of group.
- Compute powers of elements of groups, and be able to use the basic properties given by Lemma 11.3.13 and Theorem 11.3.14.

### Section 11.4: Greatest Common Divisors and the Integers Modulo $n$

- Do computations using modular arithmetic.
- $\mathbb{Z}_n$  is a group under addition and  $\mathbb{U}_n$  is a group under multiplication.  
Be able to do computations with these groups, including the group operations, exponentiation, and finding inverses.

### Section 11.5: Subsystems

- Use conditions (Theorem 11.5.3 / 11.5.5) for checking if a subset of a group is a subgroup.
- Given an element of a group, compute the cyclic group generated by that element.
- Determine if a subgroup is a cyclic group.
- Compute the order of an element in a group.

### Section 11.6: Direct Products

- Do computations involving direct products of groups, including finding the identity element, inverses, and powers of elements.
- Be able to determine if a subset of a direct product is a subgroup.

### Section 11.7: Isomorphisms

- Given two groups that are isomorphic, give an isomorphism and prove that it works using the definition.
- If two groups aren't isomorphic, explain why not.

### Section 13.1: Posets Revisited

- Given a poset, be able to draw the Hasse diagram.
- Given a poset, be able to find lower / upper bounds, greatest lower bound, least upper bound, greatest element, and least element. In particular, you should be able to compute these objects for the following examples of posets:
  - A set of integers with divides relation
  - The power set  $\mathcal{P}(S)$  with subset relation

### Section 13.2: Lattices

- Know basic terminology: Join / meet, lattice, distributive lattice
- Determine join and meet for any two elements of the lattice.
- Determine if a given poset is a lattice.

### Section 15.1: Cyclic Groups

- Know basic terminology: Cyclic group definition.
- Be able to determine if a group is cyclic or not.
- Know possible structures for a cyclic group (Theorem 15.1.9), the structure of subgroups (Theorem 15.1.10), and the order of elements in a cyclic group (Theorem 15.1.10).
- Given a cyclic group: Be able to give an isomorphism from  $\mathbb{Z}$  or  $\mathbb{Z}_n$  to the cyclic group.
- Compute the order of elements in a cyclic group.
- State the types of subgroups that could occur in terms of  $\mathbb{Z}$  or  $\mathbb{Z}_n$ .

### Section 15.2: Cosets and Factor Groups

- Know basic terminology: Left / right cosets, coset representative, factor group.
- Compute these objects for abelian groups.
- For factor groups, be able to perform the group operation, give the identity element, and compute inverses.

### Section 15.3: Permutation Groups

- Use notation for permutations: Matrix form and cycle notation.
- Express permutations as products of disjoint cycles.
- Find inverses and compositions of permutations.
- Know basic terminology: Symmetric group  $S_n$ , dihedral group  $D_n$ .
- Know how elements of  $D_n$  represent symmetries of regular  $n$ -gons. Be able to find inverses and compositions of elements in  $D_n$ , and describe the results geometrically.

### Section 15.5: Coding Theory, Group Codes

- For error correcting codes:
  - Know basic terminology: code word, generator matrix, parity-check matrix, syndrome
  - Know how to encode / decode messages.