

Section 9.7: Planar Graphs

- Understand the definition of a planar representation of a graph and be able to find a planar representation for a graph (when one exists).
- Memorize Euler's Formula, know the definition of a region in a planar representation of a graph, and be able to verify that Euler's Formula holds for a given planar graph.
- Know the corollaries to Euler's Formula and be able to use them to show that a graph is not planar.
- Know and be able to apply Kuratowski's Theorem.

Section 9.8: Graph Coloring

- Know the definitions of a vertex coloring of a graph, and the (vertex) chromatic number of a graph.
- Be able to find the (vertex) chromatic number of a given graph.
- Know the four color theorem and be able to find the dual graph of a planar map.
- Be able to use graph vertex colorings to solve applications problems (like scheduling and radio frequency assignments).
- Know the definitions of an edge coloring of a graph, and the (edge) chromatic number of a graph.
- Be able to find the edge chromatic number of a given graph.

Section 10.1: Introduction to Trees

- Understand the definition of a tree, a forest, a simple circuit.
- Understand the proof of the theorem that states that an undirected graph is a tree if and only if there is a unique simple path between any pair of its vertices.
- Understand the definition of a rooted tree, parent vertices, children, descendants, ancestors, internal vertices, leaves, siblings, an m -ary tree, a full m -ary tree, and the level of a vertex in a rooted tree.
- Understand the proof of the theorem that states that a tree with n vertices has $n - 1$ edges.
- Understand the proof of the theorem that states that a full m -ary tree has $n = mi + 1$ vertices.
- Understand the proof of the theorem that states that there are at most m^h leaves in m -ary tree of height h .

Section 10.2: Applications of Trees

- Understand the definition of a binary search tree and be able to construct a binary search tree for a given list of elements from a totally ordered set.
- Understand how to build the decision tree for a specific application and be able to use such a tree to find an algorithm to solve the given application.
- Understand the definition of a prefix code, and be able to find the Huffman binary coding given an alphabet and a set of letter frequencies.
- Given the tree of a prefix code, be able to decode binary encoded strings.

Section 10.3: Tree Traversal

- Understand how to find the universal address system for a given rooted m -ary tree.
- Be able to find the preorder, inorder, and postorder traversal for a given m -ary tree.
- Be able to draw the tree of a given computation and then give the expression for the computation in prefix, infix, and postfix form.
- Be able to draw the tree of a computation given in prefix, infix, or postfix form, and be able to compute the value of the computation.

Section 10.4: Spanning Trees

- Know the definition of a spanning tree for a connected graph and be able to find a spanning tree for a given graph.
- Understand how to find a spanning tree for a graph using a depth first search.
- Understand how to find a spanning tree for a graph using a breadth first search.
- Understand how to solve application problems using backtracking and spanning tree constructions.

Section 10.5: Minimum Spanning Trees

- Understand the definition of a minimum spanning tree for a connected weighted graph.
- Understand how to use Prim's Algorithm to find a minimum spanning tree for a given connected weighted graph.
- Understand how to use Kruskal's Algorithm to find a minimum spanning tree for a given connected weighted graph.
- Understand how to solve application problems by finding a minimum (or maximum) spanning tree.