• Identify the *vertices* and *edges* in a *graph*

• Identify if a given graph is *connected*

• Determine the *valence* of each vertex of a graph

• Use the **Euler circuit theorem** to determine whether or not a graph contains an *Euler circuit*

• *Eulerize* a graph which does not contain an Euler circuit

• Explain the difference between an Euler circuit and a *Hamiltonian circuit*.

• Calculate the number of distinct Hamiltonian circuits in a complete graph with a given number of vertices.

• Use the **brute force method of trees** to find the minimum-cost Hamiltonian circuit in a complete graph.

• Use the **nearest-neighbor algorithm** to find a candidate for the minimum-cost Hamiltonian circuit in a complete graph.

• Use the **sorted-edges algorithm** to find a candidate for the minimum-cost Hamiltonian circuit in a complete graph.

• Determine whether or not a graph is a *tree*.

• Use **Kruskal’s Algorithm** to find the *minimum-cost spanning tree* in a given weighted graph

• Find the *chromatic number* of a tree, a complete graph, and a cycle

• Find upper and lower bounds for the chromatic number of a given graph

• Convert a scheduling problem to a vertex coloring problem.

• Use the **greedy algorithm** to find a vertex coloring of a graph

• Use the **dual graph** to solve a map coloring problem

• Find the probability model for a given experiment

• Create a probability model for an unfair dice.

• State the rules of probability

• Determine whether two events are *disjoint*

• Use probabilities to determine whether two events are *independent*.

• Draw a tree diagram to model a probability experiment

• Use a tree diagram to compute basic probabilities using the rules of probability

• Use **Bayes’ Theorem** to compute conditional probability