Problem 1:

Let $G=(V,E)$ be a connected graph and $v$ be a node of $G$. The eccentricity $e(v)$ of $v$ is the distance to a node farthest from $v$. Thus:

$$e(v) = \max \{ d(u,v) : u \in V \}$$

where $d(u,v)$ is the shortest distance from $v$ to $u$.

The radius $R(G)$ is the minimum eccentricity of the nodes, whereas the diameter $D(G)$ is the maximum eccentricity. Node $v$ is a central node if $e(v) = R(G)$, it is a peripheral node if $e(v) = D(G)$.

In the following, we assume that $G$ is a weighted graph, $n = |V|$, and $e = |E|$.

1- In this part, we assume that the weights are any positive integer:
   a) Give an algorithm that finds the set of central nodes and the set of peripheral nodes in $G$.
   b) Give the time complexity of your algorithm.

2- In this part, we assume that the weights are equal:
   a) Give an $O(e)$ single source shortest path algorithm.
   b) Using the algorithm in part 1-a, write an algorithm to find the set of central nodes and the set of peripheral nodes in $G$. Analyze the time complexity of your algorithm.

Problem 2:

A greedy algorithm for graph coloring can be stated as follows: Let $A$ be the set of nodes colored so far. Initially $A$ is empty. The colors are 1,2,3,... . Choose an arbitrary node and color it "1" and put it in $A$. At any step, pick a non-colored node and color it with the lowest possible color that does not coincide with the colors of adjacent nodes. Continue this process until all nodes are colored.

(a) Give the complexity of this algorithm.
(b) Show by a counter example that this greedy method does not color graphs with minimum possible colors.

Problem 3:

We would like to run $n$ programs on $n$ different machines. Each machine will take exactly one single program. If a program $i$ runs on a machine $j$ for $1 \leq i,j \leq n$, then the cost of running this program will be $c_{ij}$. Give a strategy to minimize the total cost to run all programs.

1. State your paradigm.
   Give a pseudocode for your strategy.

Problem 4:

Use the branch and bound problem to solve the Knapsack problem.

Give an example of the search tree of your solution.