Problem 1. [20 points]
KT Chapter 4 - Problem 5. Let’s consider a long, quiet country road with houses scattered very sparsely along it. (We can picture the road as a long line segment, with an eastern endpoint and a western endpoint.) Further, let’s suppose that despite the bucolic setting, the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations.

Give an efficient algorithm that achieves this goal, using the minimum number of base stations.

Problem 2. [Proof problem - 5 points] Prove that the algorithm you defined for the problem above is correct.

Problem 3. [20 points]
KT Chapter 3 Problem 6. We have a connected graph \( G = (V, E) \), and a specific vertex \( u \in V \). Suppose we compute a depth-first search tree rooted at \( u \), and obtain a tree \( T \) that includes all nodes of \( G \). Suppose we then compute a breadth-first search tree rooted at \( u \), and obtain the same tree \( T \). Prove that \( G = T \).

(Hint: if \( T \) is both a depth-first search tree and a breadth-first search tree rooted at \( u \), then \( G \) cannot contain any edges that do not belong to \( T \).)

Problem 4. [20 points]
Suppose that you have an undirected graph representing a city’s transportation network. Each edge represents a street (which we’ll assume is a two-way street), and each node represents an intersection. Certain intersections in the city are hospitals, and you are interested in finding, for each intersection, the distance that intersection is from the nearest hospital, as measured by the number of edges in the path from that intersection to the nearest hospital.

Let \( n \) be the number of nodes in the transportation network, let \( m \) be the number of edges, and let \( k \) be the number of nodes in the grid that are hospitals. Assuming that you know the \( k \) nodes that are hospitals,

(a) Design an algorithm for computing the distance from each intersection to the closest hospital.
(b) Describe your algorithm and its run time analysis.
(c) Does your run time depend on the total number of hospitals? If yes, explain how you can modify the algorithm, and make its asymptotic run time not depend on the number of hospitals.

Problem 5. [20 points - Peer review allowed]
KT Chapter 4 - Problem 3. You are consulting for a trucking company that does a large amount of business shipping packages between New York and Boston. The volume is high enough that they have to send a number of trucks each day between the two locations. Trucks have a fixed limit \( W \) on the maximum amount of weight they are allowed to carry. Boxes arrive at the New York station one by one, and each package \( i \) has a weight \( w_i \). The trucking station is quite small, so at most one truck can be at the station at any time.

Company policy requires that boxes are shipped in the order they arrive; otherwise, a customer might get upset upon seeing a box that arrived after his make it to Boston faster. At the moment, the company is using a simple greedy algorithm for packing: they pack boxes in the order they arrive, and whenever the
next box does not fit, they send the truck on its way.

But they wonder if they might be using too many trucks, and they want your opinion on whether the situation can be improved. Here is how they are thinking. Maybe one could decrease the number of trucks needed by sometimes sending off a truck that was less full, and in this way allow the next few trucks to be better packed.

(a) What is the greedy algorithm proposed in this problem?
(b) To establish the optimality of this greedy packing algorithm, you need to identify a measure under which it stays ahead of all other solutions. What is that measure?
(c) Prove that, for a given set of boxes with specified weights, the greedy algorithm currently in use actually minimizes the number of trucks that are needed. Your proof should follow the type of analysis we used for the Interval Scheduling Problem.

Problem 6. [15 points]
Remember your algorithm from last assignment to find a cycle in a graph with complexity of $O(n+m)$. In this problem, give an algorithm that determines whether or not a given undirected graph $G = (V, E)$ contains a cycle. Your algorithm should run in $O(n)$ time only, independent of $m$.

(a) Provide pseudocode for your algorithm.
(b) Justify your algorithm’s running time.