Problem 1. [10 points]
A binary tree is a rooted tree in which each node has at most two children. Show by induction that in any binary tree the number of nodes with two children is exactly one less than the number of leaves.

Problem 2. [10 points]
Let’s consider a long, quiet country road with house 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.

(a) Give an efficient algorithm that achieves this goal, using the minimum number of base stations.
(b) Show that your algorithm is correct (no need for a formal proof, but justify its correctness clearly).

Problem 3. [10 points - Proof problem]
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 4. [10 points]
We have a connected graph G = (V, E), and a specific vertex u ∈ 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 5.  [10 points]

You’re helping a group of ethnographers analyze some oral history data they’ve collected by interviewing
members of a village to learn about the lives of people who’ve lived there over the past two hundred years.

From these interviews, they’ve learned about a set of $n$ people (all of them now deceased), whom we’ll denote
$P_1, P_2, ..., P_n$. They’ve also collected facts about when these people lived relative to one another.

Each fact has one of the following two forms:
* For some $i$ and $j$, person $P_i$ died before person $P_j$ was born; or
* for some $i$ and $j$, the life spans of $P_i$ and $P_j$ overlapped at least partially.

Naturally, they are not sure that all these facts are correct; memories are not so good, and a lot of this was
passed down by word of mouth. So what they’d like you to determine is whether the data they’ve collected
is at least internally consistent, in the sense that there could have existed a set of people for which all the
facts they’ve learned simultaneously hold.

Give an efficient algorithm to do this: either it should produce proposed dates of birth and death for each of
the $n$ people so that all the facts hold true, or it should report (correctly) that no such dates can exist that
is, the facts collected by the ethnographers are not internally consistent.

You don’t need to write complete pseudocode for this algorithm, just explain it as clearly as you can.