MATH 113: DISCRETE STRUCTURES
WEDNESDAY WEEK 7 HANDOUT

Problem 1. Let $G = (V, E)$ be a graph with connected subgraphs $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ such that $V_1 \cap V_2 \neq \varnothing$. Prove that G is connected.

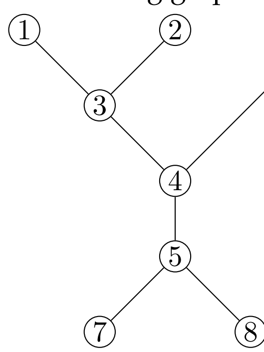
Call a graph *acyclic* if it does not contain any subgraphs which are cycles. A *tree* is a connected acyclic graph. A disconnected acyclic graph is called a *forest*.

Problem 2. How many edges are there in a tree with n vertices? Prove your assertion (by induction?).

Problem 3. Prove that a graph G is a tree if and only if there is a unique path between any two vertices of G.

A *leaf* of a tree is a vertex of degree 1. Suppose T is a tree with vertex set $\underline{n} = \{1, 2, \ldots, n\}$. The *Prüfer code* of T is the sequence of length $n - 2$ with entries in $\underline{n}$ generated by the following algorithm: At step i, remove the leaf with the smallest label and set the i-th entry of the Prüfer sequence equal to the label of the leaf's neighbor. After step $n - 2$, the end of the algorithm, one is left with a single edge joining two nodes.

For instance, the Prüfer code of the following graph is $(3, 3, 4, 4, 5, 5)$.



Problem 4. Draw several trees on the whiteboards and determine their Prüfer codes.

Problem 5. Create an algorithm (*i.e.*, sequence of instructions) which turns a Prüfer code in $\underline{n}$ of length $n - 2$ into a tree with vertex set $\underline{n}$ such that the Prüfer code $\to$ tree and tree $\to$ Prüfer code processes are inverses. Conclude that there are n^{n-2} trees with vertex set $\underline{n}$. This is called *Cayley's Theorem*.