Problem 1. *Find all possible graphs with the given degree sequence or prove that none exists. In either case, show your work.*

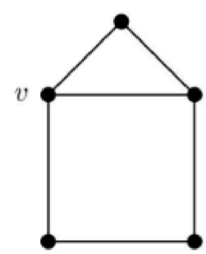
(a) $(3, 3, 2, 2, 2)$

Solution. First, we can use the *Havel-Hakimi Theorem* to prove that the degree sequence is graphic:

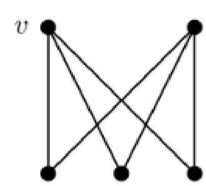
$$
\begin{aligned}
(3,3,2,2,2) &\to (2,1,1,2) && \text{removed 3 edges} \\
(2,2,1,1) &\to (1,0,1) && \text{removed 2 edges} \\
(1,1,0) &\to (0,0) && \text{removed 1 edge.}
\end{aligned}
$$

Because $(0, 0)$ is clearly graphic (it's just two points with no edges), we know that there must be some graph that satisfies the degree sequence $(3, 3, 2, 2, 2)$. Moreover, such a graph must have 6 edges. Let v be a vertex with degree 3. There are two possibilities based on this information:

- **Case 1 (v is adjacent to vertices of degree $(3, 2, 2)$):** Once the two vertices of degree 3 are connected to each other, the only way to maintain this degree sequence is for the other vertex of degree 3 to connect to the lone vertex that is not adjacent to v (if this didn't happen, then have a graph with 5 edges and 5 vertices with all correct degrees; there'd be no way to give the remaining vertex degree 2 with the one edge we have left). Thus, the unique isomorphism class this case creates is the *house*.



- **Case 2 (v is adjacent to vertices of degree $(2, 2, 2)$):** This requires the other vertex of degree 3 to also be connected to the three vertices of degree 2. The unique isomorphism class this case creates is $K_{2,3}$.



There is no other way to define the neighborhood of v, so this list must be exhaustive. □

(b) $(5, 5, 4, 4, 2, 2)$

Solution. There is no graph that satisfies this degree sequence. Consider the following application of the *Havel-Hakimi Theorem*:

$$
\begin{aligned}
(5,5,4,4,2,2) &\to (4,3,3,1,1) && \text{removed 5 edges} \\
(4,3,3,1,1) &\to (2,2,0,0) && \text{removed 4 edges}
\end{aligned}
$$

But, $(2, 2, 0, 0)$ is not graphic (the leading vertex with degree 2 can only connect to the other vertex of degree 2; it's impossible to get past degree 1 with a simple graph). Therefore, no graph exists with the degree sequence $(5, 5, 4, 4, 2, 2)$. $\square$

Problem 2. *Determine which trees have Prüfer codes that*

(a) contain only one value;

(b) contain exactly two values;

(c) have distinct values.

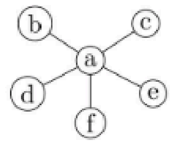
Be sure to explain your answer.

Solution. (a) **Only one value.** If $P(T) = (a, a, \ldots, a)$, then a appears $n-2$ times, so

$$\deg(a) - 1 = n - 2 \Rightarrow \deg(a) = n - 1.$$

Every other vertex appears 0 times, hence has degree 1. Therefore T is exactly the **star** centered at a.

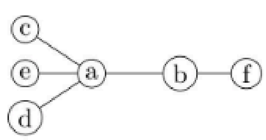
Example (star of 6 vertices centered on a):



(b) **Exactly two values.** Suppose $P(T)$ uses exactly two labels a and b (each at least once). Then every other vertex appears 0 times, so has degree 1 (is a leaf). Thus only a and b can have degree ≥ 2. Additionally, a and b must be adjacent. If not, the unique path from a to b would pass through some other vertex of degree ≥ 2, which would then appear in the Prüfer code thus giving a contradiction contradiction.

So T is an edge ab with some number leaves attached to a and the remaining leaves attached to b.

Example :



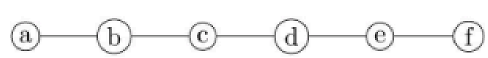
Here a has degree 4, b has degree 2; Prüfer code uses only a and b.

(c) **All distinct values.** If all entries of $P(T)$ are distinct, then each vertex appears either 0 or 1 times. Hence each vertex has

$$\deg(v) = 1 \quad \text{(if it appears 0 times)}, \qquad \deg(v) = 2 \quad \text{(if it appears 1 time)}.$$

So every vertex has degree ≤ 2. A connected graph with maximum degree 2 and two leaves is a path. In a path the two endpoints have degree 1 (appear 0 times in P(T) and the other $n-2$ vertices have degree 2 (appear once), so the Prüfer code has no repeats.

Example (path on 6 vertices):



$\square$

Problem 3. *Use Prufer codes and Cayley's theorem to prove that the graph obtained from K_n by deleting an edge has $(n-2)n^{n-3}$ spanning trees.*[1]

Solution. By Cayley's Theorem, given n labeled vertices, we know that there are n^{n-2} different trees that we can create. We could rephrase this as saying, given the complete graph K_n, there are n^{n-2} different spanning trees. Now, suppose we remove an edge $\{n, n-1\}$ from the graph K_n. Let's call this new graph G. Note that we could have removed any edge from the graph K_n since they are all isomorphic to one another.

To determine how many spanning trees G has, all we have to do is start with the n^{n-2} spanning trees of the graph K_n and remove all the spanning trees that contain that edge $\{n, n-1\}$. To do this, take a given spanning tree T_i of K_n with a Prüfer code $(a_1, \ldots, a_{n-2})$ such that $a_i \in \{1, \ldots, n\}$. Since n and $n-1$ are the largest vertices of T_i, in the creation of the Prüfer code of T_i, all other leaves of the tree are removed before them. Therefore, if n and $n-1$ form an edge in T_i, then n or $n-1$ must be the last value of the Prüfer code of T_i. This is because if $\{n, n-1\}$ is in the tree, it must, by construction, be the last edge left in the graph during the construction of the Prüfer code, meaning that the last leaf that was removed from the tree had a neighbor of either n or $n-1$.

Thus, the Prüfer code of T_i must end in n or $n-1$ for $\{n, n-1\}$ to be in T_i. If the Prüfer code of T_i ends in n, there are n^{n-3} possible Prüfer codes for T_i as the other $n-3$ slots in the code can be one of n values. Thus, there are n^{n-3} spanning trees T_i that have a Prüfer code ending in n and by symmetry, there are n^{n-3} spanning trees T_i that have a Prüfer code ending in $n-1$.

Thus, there are $2n^{n-3}$ spanning trees T_i have a Prüfer code ending in n or $n-1$. As a result, there are $2n^{n-3}$ spanning trees of K_n including the edge $\{n, n-1\}$ and $n^{n-2} - 2n^{n-3} = (n-2)n^{n-3}$ spanning trees of K_n that do not include the edge $\{n, n-1\}$. Thus, the graph G has $(n-2)n^{n-3}$ spanning trees. □

[1] Remark/hint: One way to count how many fingers are on your right hand is to first count how many fingers you have in total and subtract the number of fingers on your left hand...

Problem 4. *Prove that the number of labeled n-vertex graphs where every vertex has even degree is* $2^{\binom{n-1}{2}}$. [2]

Solution. Given a labeled graph H with $n-1$ vertices $\{1, 2, \ldots, n-1\}$. Define a graph G with vertices $\{1, 2, \ldots, n\}$ by

$$E(G) = E(H) \cup \{\{i, n\} : i \text{ has odd degree in } H\}$$

Then every vertex in G has even degree. If i has even degree in H, no new edges incident to i are added, so i has even degree in G; if i has odd degree in H, one new edge incident to i is added, to i has even degree; since there are always an even number of odd-degree vertices in a graph, n has even degree.

This construction is certainly injective. It is surjective too. Given a graph G with n labeled vertices all with even degree, remove the vertex n and all edges incident to it. The resulting graph H is such that the construction above yields G. Thus the number of labeled n-vertex graphs with all vertices of even degree is the same as the number of $(n-1)$-vertex graphs, which is $2^{\binom{n-1}{2}}$. $\square$

[2]Hint: establish a bijection to the set of all graphs with labeled $(n-1)$-vertex graphs.

Problem 5. *Consider the alternative version of Bridg-it where the player that connects their end-lines* loses. *Use spanning trees to show that Player* 2 *has a winning strategy in this game.*

Solution. We can represent the Bridg-it game as the union of two complementary spanning trees T and T' with an additional edge from connecting both ends of the board. Now we want to show that for any edge from T' added to T which makes T there exists an edge of T which we can remove to make it a spanning tree again. Since T is a spanning then there exists a unique path from any vertices u and v. Then suppose we add $e' \in T'$ to T. Then this is a new edge which creates a new connection between u and v. This closes the path, creating a new cycle. Since this is the only cycle we can choose any edge $e \in T$ such that $e \neq e'$ to make T a spanning tree again. Now observe the cases when player 2 must delete the added edge to maintain the spanning tree and when they can choose another edge to delete which maintains the spanning tree.
Case 1. Suppose player 2 must delete the edge connecting the two end lines. Then this added edge is part of a cycle and so there exists another path between the end lines where player 1 has placed their pieces. Hence player 1 lost.
Case 2. Then player 2 can choose an edge to remove such that player 1 is forced to maintain a spanning tree.
Therefore player 2 has a winning strategy by forcing player 1 to maintain a spanning tree which will result in player making a move that connect both end lines. □

Problem 6. *Prove that if $T_1, \dots, T_k$ are pairwise-intersecting subtrees of a tree T, then some vertex of T belongs to each of $T_1, \dots, T_k$.* [3] [4]

Solution We prove the statement by induction on k. For $k = 1$ the claim is trivial. For $k = 2$, since T_1 and T_2 intersect, any vertex in $V(T_1) \cap V(T_2)$ belongs to both subtrees, so the claim holds. Assume the statement holds for $k - 1 \geq 2$ pairwise-intersecting subtrees, and let $T_1, \dots, T_k$ be pairwise-intersecting subtrees of a tree T. By the induction hypothesis applied to $T_1, \dots, T_{k-1}$, there exists a vertex $v \in \bigcap_{i=1}^{k-1} V(T_i)$. If $v \in V(T_k)$, then $v \in \bigcap_{i=1}^{k} V(T_i)$ and we are done. Assume therefore that $v \notin V(T_k)$. For each $i = 1, \dots, k-1$, choose a vertex $x_i \in V(T_i) \cap V(T_k)$, which exists since the subtrees are pairwise intersecting. Let P_i be the unique path in T from v to x_i. Since $v, x_i \in T_i$ and T_i is connected, the entire path P_i lies in T_i. For each i, let y_i be the first vertex on P_i, when traveling from v toward x_i, that lies in T_k. Then $y_i \in V(T_i) \cap V(T_k)$. Choose one of these vertices, say $y = y_j$, such that the distance from v to y is minimal among all $y_1, \dots, y_{k-1}$. We will claim that $y \in V(T_i)$ for every $i = 1, \dots, k-1$. Without loss of generality we will fix i. Consider the paths P_i and P_j (where P_j is the unique oath in T from v to x_j). Since T is a tree, these paths share a common initial segment starting at v, and then diverge at some vertex w. By definition of $y = y_j$, no vertex on the path from v to y lies in T_k. In particular, the vertex w, which lies on this initial segment, is not in T_k. Since $x_i \in T_k$, the path P_i must enter T_k at some point after w, and by definition this first entry point is y_i. If the path P_i entered T_k at a vertex closer to v than y, this would contradict the minimality of the choice of y. Therefore, the first point where P_j enters T_k, which is y, must lie on the path P_i. Since $P_i \subseteq T_i$, it follows that $y \in V(T_i)$. Thus $y \in V(T_i)$ for all $i = 1, \dots, k-1$, and by construction $y \in V(T_k)$. Hence $y \in \bigcap_{i=1}^{k} V(T_i)$, which completes the induction step. $\square$

[3] Remark: This is a graph-theoretic analog of Helly's theorem.

[4] Hint: use induction on k.