1. Consider a directed graph.

An Eulerian walk is a walk that goes through each node and edge exactly once, and an Eulerian circuit is an Eulerian walk that starts and ends at the same node. Assume that the indegree plus outdegree of every node is nonzero.

Prove or Disprove: In a given directed graph, if an Eulerian circuit exists, then it is strongly connected.

True. Let G be a directed graph with an Eulerian circuit C. For any two distinct vertices x, y ∈ V(G),

- Since C visits every vertex, it passes through x at some point.
- Starting from x and following C until we reach y yields a directed walk from x to y.

Thus, there exists a directed path from x to y, which means that G is strongly connected because we picked x and y arbitrarily.

2. Prove that a directed graph with n >= 2 nodes has an Eulerian circuit if and only if it is strongly connected, every node has nonzero indegree or outdegree, and each node's indegree is equal to its outdegree.

We prove both sides of the statement:

[=>] "If G has an Eulerian circuit C, then…

- it is strongly connected: for any u, v ∈ V(G), follow C from u to v to get a directed walk, hence a directed path. Therefore, G is strongly connected.
- nonzero indegree/outdegree: every vertex is visited by C, so each vertex has an indegree at least one, and an outdegree at least one since there should be an edge entering/leaving each vertex
- indegree = outdegree for each node: each time C enters a vertex via an edge, it leaves via another edge. thus, each visit contributes 1 to both the indegree and the outdegree, and every edge is counted exactly once, hence the total indegree is equal to the total outdegree for each vertex.

[<=] "If G is strongly connected, every vertex has nonzero indegree or outdegree and indegree = outdegree for all v, then G has an Eulerian circuit."

We show G has an Eulerian circuit via Hierholzer's algorithm:

1. Start at any vertex x.
2. Follow unused edges arbitrarily until returning to x, yielding a closed walk W.

3. If unused edges remain, pick a vertex on W with unused edges and repeat, inserting the new closed walk into W.
4. Continue until all edges have been used.

This is correct because indegree = outdegree guarantees that whenever we enter a vertex except at the start, there is an unused outgoing edge, so we never get stuck except at the start. Strong connectivity ensures we can reach any remaining edges. The process terminates after all edges have been used, producing an Eulerian circuit.

3. For fixed n >= 0, let G_n be the directed graph with:

- **Vertices: all 2^n binary strings of length n over {a,b}**
- **Edges: for each binary string s = $s_0s_1 \ldots s_n$ of length n + 1, add a directed edge from $s_0s_1 \ldots s_{n-1}$ to $s_1s_2 \ldots s_n$.**
- **Prove that G_n has an Eulerian circuit for all n >= 0.**

We use the three conditions from Problem 2:

1. Indegree = Outdegree for every vertex
    a. Every node has outdegree n, since for each string s, the strings s*a* and s*b* determine all outgoing edges from s.
    b. Every node has indegree n, since for each string s, the strings *a*s and *b*s determine all outgoing edges from s.
    c. The previous two points imply that every node's indegree = outdegree.
2. Note that the graph is strongly connected, that for each pair of strings s and t, one can go from s to t by repeatedly going through the edge corresponding to each letter of t in oder.

Therefore, the graph has an Eulerian circuit.

4. Consider an embedding of a connected undirected planar graph with $n > 0$ nodes. Call the graph G and its dual $\overline{G}$.

   Recall that the nodes of $\overline{G}$ corresponds to faces, and each edge of G corresponds to an edge in $\overline{G}$ that connects the faces on its two sides. In particular, G and $\overline{G}$ have the same number of edges. For each edge e in G, denote this corresponding edge in $\overline{G}$ by $\overline{e}$.

   Let H and H' be subgraphs of G and $\overline{G}$, respectively, such that:

   - H contains all the nodes of G,
   - H' contains all the nodes of $\overline{G}$,
   - For each edge e in G, either e is in H or $\overline{e}$ is in H', but not both. (In particular, the total number of edges in H and H' combined is equal to the number of edges in G.)

4.1. Prove that if H is connected, then H' is acyclic.

We prove the contrapositive: "If H' has a cycle, then H is not connected."

Consider a cycle of H'. Since H' is a subgraph of the dual G_, this cycle consists of faces.

Since the vertices of H inside it are disconnected from the faces outside it, the only way to "get out" is through some edge that joins two faces in the cycle, but since those edges are all in H', they are not in H. Therefore, H is not connected.

4.2. Prove that if H is acyclic, then H' is connected.

We prove the contrapositive: "If H' is disconnected, then H has a cycle." Consider the outermost face of H'. Since H' is disconnected, there is some face that it cannot reach. Coloring all faces that it cannot reach, the boundary of the colored faces constitute a cycle in H.

4.3. Prove that H is a tree iff H' is a tree.

[=>] If H is a tree, then H is connected and acyclic, using 4.1 and 4.2, H' would also be connected and acyclic, so H' is a tree.

[<=] If H' is a tree, performing the same construction as above on the dual using the previous result, H'' is also a tree, but since H'' is isomorphic to H, it proves that H is also a tree.

4.4. Prove that G has the same number of spanning trees as G_.

The previous result shows that the construction is a one-to-one correspondence between spanning trees of G to its dual, so they have the same number of spanning trees.