Instructions: Do all problems and typeset them in LaTeX. E-mail the PDF file to Jeremy at jlmartin@ku.edu under the filename `your-last-name.pdf` by **Friday, February 12, 5:00pm.** You are encouraged to use the header file at http://www.math.ku.edu/~jmartin/math725/header.tex.

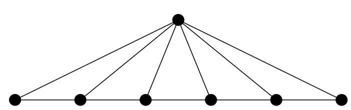
Problem #1 Let $G = (V, E)$ be a connected graph, let $T, T' \in \mathscr{T}(G)$, and let $e \in T \backslash T'$. Prove that there exists an edge $e' \in T' \backslash T$ such that both $T - e + e'$ and $T' + e - e'$ are spanning trees. (This is known as the "symmetric exchange law.")

Problem #2 **(#2a)** Let $k \geqslant 2$ and let G be a k-regular bipartite graph. Prove that G has no cut-edge. (Hint: Use the bipartite version of handshaking.)

(#2b) Construct a simple, connected, nonbipartite 3-regular graph with a cut-edge. (This shows that the condition "bipartite" really is necessary in (a).)

Problem #3 Let L_n be the graph obtained from K_n by deleting one edge. Determine $\tau(L_n)$. (Hint: Use Cayley's formula as a starting point.)

Problem #4 Let F_n be the graph obtained from P_n by adding a new vertex adjacent to every vertex of P_n. For example, $F_1 \cong P_2$, $F_2 \cong K_3$, and $F_3 \cong L_4$ (as in the previous problem), and F_6 is shown below.



Let $a_n = \tau(F_n)$. Use deletion/contraction to prove that $a_n = 3a_{n-1} - a_{n-2}$ for $n \geqslant 3$. See if you can recognize the sequence $a_1, a_2, a_3, a_4 \ldots$

Problem #5 Let $K_{p,q}$ denote the complete bipartite graph with partite sets of sizes p and q. Use the Matrix-Tree Theorem to calculate $\tau(K_{p,q})$. (Suggestion: Find an explicit basis for $\mathbb{R}^{p+q}$ consisting of eigenvectors of the Laplacian matrix $L(K_{p,q})$.)

Problem #6 Let G be a connected graph with weight function $w : E(G) \to \mathbb{R}_{\geqslant 0}$.

(#6a) Suppose that $C \subseteq G$ is a cycle and $e \in C$ is an edge of maximum weight (i.e., $w(e) \geqslant w(e')$ for all $e' \in C$). Prove that G has an MST not containing e.

(#6b) Use (a) to show that the following algorithm produces an MST for all G and w:

```
Let T := G
while T contains a cycle do:
    Let C be a cycle
    Let e be an edge of C of maximum weight
    Set T := T − e
Return T
```