1, Diestel 1.31: Prove or disprove that a graph is bipartite if and only if no two adjacent vertices have the same distance from any other vertex.

Proposition 1.6.1 in Diestel states that a graph is bipartite if and only if it contains no odd cycle, so it suffices to show that a graph has an odd cycle if and only if it has a pair of adjacent vertices that are the same distance from another vertex.

($\Rightarrow$) Let G be a graph with an odd cycle. Let $C = x_0x_1 \cdots x_m x_0$ be a minimum odd cycle in G. (Note that C has size $m+1$.) We prove a lemma that for $x, y \in C$, $d_G(x,y) = d_C(x,y)$. WLOG let $x = x_0$ and let $y = x_i$ and $i \leq \frac{m+1}{2}$ (reindexing the cycle in the other direction handles the case where i is larger). We know $d_C(x,y) = i$, so suppose $d_G(x,y) < i$. Then there is an x–y path P of length less than i. Because P starts and ends in P, but is shorter than $x_0x_1 \cdots x_i$, P contains a subpath P' with endpoints x_j and x_k that is a C–path and is shorter than $d_C(x_j, x_k)$. Consider two cycles:

$$C_1 = x_j x_{j+1} \cdots x_k P x_j$$

and

$$C_2 = x_j P x_k x_{k+1} \cdots x_m x_0 x_1 \cdots x_j.$$

Let l be the length of P'. Then C_1 has size $k - j + l$ and C_2 has size $(m+1) - (k-j) + l$. Because $l < (k-j) \leq i \leq \frac{m+1}{2}$, C_1 and C_2 each have size strictly less than m. Adding the lengths of C_1 and C_2 gives $(m+1) + 2l$, which is odd because $m+1$ is odd. Therefore, one of the cycles is odd (the sum of two even things is even). So, we have found a smaller odd cycle, which is a contradiction. We conclude that for $x, y \in C$,

$$d_G(x,y) = d_C(x,y).$$

At this point, it suffices to take x_0, x_1, and $x_{(m+1)/2}$ and observe that x_0 and x_1 are adjacent and

$$d_G(x_1, x_{(m+1)/2}) = d_C(x_1, x_{(m+1)/2}) = \frac{m-1}{2} = d_C(x_0, x_{(m+1)/2}) = d_G(x_0, x_{(m+1)/2}).$$

($\Leftarrow$) Let G be a graph with vertices x, y, z such that $xy \in E(G)$ and

$$d(x,z) = d(y,z).$$

We do not consider infinite distances to be equal, so these distances are finite. Let P_1 and P_2 be x–z and y–z paths of minimum length (respectively). Let m be the first vertex in P_1 that is also in P_2 (such a vertex exists because z is in both paths). We claim xP_1m and yP_2m must have the same length. If WLOG $|xP_1m| < |yP_2m|$ then $|mP_1z| > |mP_2z|$ and $xP_1m + mP_2z$ is a shorter x–z path than P_1, and this is a contradiction. We also know that x does not appear in P_2 and y does not appear in P_1 by a similar argument: xP_2z would be a shorter x–z path than P_1, and yP_1z would be a shorter y–z path than P_2, so $m \neq x$ and $m \neq y$.

By our choice of m, $xP_1\mathring{m}$ and $yP_2\mathring{m}$ are disjoint and nontrivial, so $C = xP_1mP_2yx$ is a cycle. We know (where $|P|$ denotes length):

$$|C| = |xP_1m| + |mP_2y| + 1 = 2|xP_1m| + 1$$

so C is an odd cycle.

2, Diestel 1.35: Prove or disprove that every connected graph contains a walk that traverses each of its edges exactly once in each direction.

Let G be an arbitrary connected graph. Given a walk, W, we will say that the walk has Property Q if the walk does not traverse an edge more than once in the same direction. Take a maximum walk with Property Q in G and call it $W = x_0 x_1 \cdots x_{k-1} x_k$. Also let $u = x_0$ and $v = x_k$

W must be closed. If W is not closed, then we observe that at vertex v, there are $\deg(v)$ edges, and so $2\deg(v)$ "ways" to traverse an edge incident with v (where a "way" is an edge plus a direction). Each time that W traverses an edge to v, xv, the next edge must be from v: vy, with the obvious exception of the last edge, $x_{k-1}x_k$. It follows that W has taken i edges to v for some positive i, and $i-1$ from v. Therefore, there there is an edge that has not been taken from v, say vz, and we extend W by adding z to the end of the sequence of vertices to form W', a walk with Property Q that is longer than W, proving that W must be closed.

W must traverse each edge twice. Suppose some edge e is not traversed twice. Because G is connected, there is an e–$V(W)$ path (this is a path starting at a vertex of e, ending at a vertex of W, and internally disjoint from e and W. If this is a trivial path, let x be the edge of e that is in W and let y be the other edge of e. Otherwise, call the end of the path x and the penultimate vertex y, and observe that y is not in W so xy is not traversed in either direction by W. In either case, x is in W and xy is not traversed in some direction by W. We reindex W by starting at x: since $x = x_i$ for some i, $W' = x_i x_{i+1} \cdots x_k x_1 \cdots x_{i-1} x_i$ is a walk with Property Q because W was a closed walk with Property Q. If xy has not been taken from x to y, then adding xy to the end of W' forms a walk longer than W that still has property Q. Otherwise, xy has not been taken from y to x, so adding xy to the end of W' forms a longer walk than W that still has property Q. In any case, we get a contradiction, so we conclude that W traverses each edge twice, and by the definition of Property Q, this means W traverses each edge once in each direction.

3, Diestel 1.49: Let $A = (a_{ij})_{n\times n}$ be the adjacency matrix of the graph G. Show that the matrix $A^k = (a'_{ij})_{n\times n}$ displays, for all $i, j \leq n$, the number a'_{ij} of walks of length k from v_i to v_j in G.

The proof goes by induction on k.

We consider two bases cases.

When $k = 0$, we see A^k is the identity matrix. Fittingly, there is exactly one walk of length 0 from any vertex to itself, and there are no walks of length zero to any other vertex.

When $k = 1$, a_{ij} is 1 exactly when vertices i and j are adjacent; that is, when there is a walk of length 1 from i to j.

Assume the claim holds for all $j < k$ for some fixed k.

Consider $A^k = A^{k-1} \times A$. Let $B = (b_{ij}) = A^{k-1}$. The i, j entry in A^k is, by definition,

$$\sum_k b_{ik} a_{kj}.$$

By our induction hypothesis, b_{ik} is the number of walks from i to k. Then, if k is adjacent to i, there are b_{ik} i–j walks with penultimate vertex k; otherwise, there are no such walks. The i, j entry of A^k, then, is the sum across all k of the number of walks with penultimate vertex k. This completes the proof.

4: Let G be a connected graph whose edges have been assigned real numbers. As mentioned on page 14 of the text, G has at least one spanning tree. The *weight* of a spanning tree is the sum of the numbers on its edges. The *spectrum* of a spanning tree is the list of the numbers on its edges (each number listed as many times as it occurs on the edges of the tree) in non-decreasing order. Show that any two spanning trees of minimum weight (among all spanning trees of G) must have the same spectrum.

We require a lemma:

Lemma 1: If T is a tree and $u, v \in V(T)$ such that $\{u, v\} \notin E(T)$, then $T' = T + \{u, v\}$ has a cycle $C = c_0, c_1, \ldots, c_m, c_0$. Furthermore, if T'' is the tree resulting from removing any edge $e = \{c_i, c_{i+1}\}$ from C, then T'' is a tree.

Proof: The first part of the lemma follows directly from Theorem 1.5.1 in Diestel. Thus all we must show is T'' is a tree. Let x and y be vertices in T''. As $V(T'') = V(T)$ and T is a tree, x and y are connected in T by a path $P : x = p_0, p_1, \ldots, p_{\ell-1}, p_\ell = y$. If $\{p_{j-1}, p_j\} \neq e$ (the edge we removed from C) for any $1 \leq j \leq \ell$, then P is a path in T''. Otherwise, suppose $\{p_{j-1}, p_j\} = e$. Then $p_{j-1} = c_i$ for some i (and $p_j = c_{i+1}$) because the edge we removed was on the cycle. So

$$P' = x = p_0, \ldots, p_{j-1} = c_i, c_{i-1}, \ldots, c_0, c_m, \ldots, c_{i+1} = p_j, p_{j+2}, \ldots, p_\ell = y$$

is an x–y walk in T''. As every walk contains a path, we conclude x and y are connected in T''. Furthermore, $|V(T'')| = |V(T)| \overset{\text{Corollary 1.5.3}}{=} |E(T)| + 1 = |E(T'')| + 1$, so by Corollary 1.5.3, T'' is a tree.

Now we prove the theorem.

Let G be a graph such that there exist minimum weight spanning trees in G that have different spectra (we will derive a contradiction from this fact). Let T_X and T_Y be two minimum weight spanning trees of G with different spectra such that (among all pairs of MSTs with different spectra):

- the lightest edge in which they differ has maximum weight
- and among all such pairs, there are as few differences in edges of that weight as possible.

T_X and T_Y are well-defined because there are finitely many spanning trees of a finite graph G, and we assumed there are spanning trees of different spectra. Differing in edge e means that one tree has e while the other does not.

Let e be one such minimum weight edge in which T_X and T_Y differ. WLOG $e \in E(T_X)$. By the lemma, $T_Y + e$ has some cycle C. Because T_X is acyclic, there must be some other edge $f \in C$ such that $f \notin T_X$. First, because T_Y is minimal, $w(f) \leq w(e)$, as otherwise $T_Y + e - f$ is a spanning tree of less weight than the minimum. Also, because e is the lightest edge in which T_X and T_Y differ, $w(f) \geq w(e)$ (because otherwise f would be the lightest edge where T_X and T_Y differ). So,

$$w(e) = w(f),$$

and again by the lemma $T_Z = T_Y + e - f$ is a minimum weight spanning tree. But, T_Z and T_X either no longer differ in edges of weight $\leq w(e)$ or now differ in fewer edges of that weight (since we removed one pair of differing edges, and the rest stayed fixed, so we have derived a contradiction.

5: An oriented complete graph is called a *tournament*. The *outdegree* of a vertex v, written $od(v)$, is the number of edges directed away from v. Let T be a tournament with n vertices. Find a formula for the number of directed 3-cycles in T in terms of n and the outdegrees of the vertices of T.

Let T be a tournament with n vertices. A vertex v of T is contained, as a vertex of outdegree two, in exactly $\binom{od(v)}{2}$ 3-vertex subtournaments which are not cycles. Each 3-vertex subtournament which is not a cycle contains exactly one vertex of outdegree two (within the subtournament itself). Thus, the number of 3-vertex subtournaments which are not cycles is $\sum_{v \in T} \binom{od(v)}{2}$. Hence, the number of 3-cycles in T is $\binom{n}{3} - \sum_{v \in T} \binom{od(v)}{2}$.

6: Let G be a graph with vertices $\{v_1, v_2, \ldots, v_n\}$. Let the matrix M be defined by

$$m_{ij} = \begin{cases} d(v_i) & \text{if } i = j \\ -1 & \text{if } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases}$$

The **Matrix Tree Theorem** states that the number of spanning trees of G is equal to the value of any cofactor of M. Use the matrix tree theorem to find the number of spanning trees in $K_{n,n}$.

[I_n is an $n \times n$ identity matrix. $J_{m,n}$ is an $m \times n$ matrix with entry all one. $0_{m,n} = 0J_{m,n}$.]

For $G = K_{n,n}$ (and $n \geq 2$),

$$M = \begin{bmatrix} n & \cdots & 0 & -1 & \cdots & -1 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & n & -1 & \cdots & -1 \\ -1 & \cdots & -1 & n & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ -1 & \cdots & -1 & 0 & \cdots & n \end{bmatrix} = \left[\begin{array}{c|c} nI_n & -J_{n,n-1} \\ \hline -J_{n-1,n} & nI_{n-1} \end{array}\right]$$

Add all of the rows except the first to the first (these are elementary row operations, and do not change the determinant). The first row is now

$$\begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix}$$

Adding the first row to each row of the bottom "half" gives

$$\det(M) = \det\left(\left[\begin{array}{c|ccc|ccc} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ \hline 0 & & & & & & \\ \vdots & & nI_{n-1} & & & -J_{n-1,n-1} & \\ 0 & & & & & & \\ \hline 0 & & & & & & \\ \vdots & & 0_{n-1,n} & & & nI_{n-1} & \\ 0 & & & & & & \end{array}\right]\right)$$

This is upper-triangular, so we multiply the entries on the diagonal to get

$$\det(M) = 1 \cdot n^{n-1} \cdot n^{n-1} = n^{2n-2}.$$

We note that there is one spanning tree on $K_{1,1}$, and so we have by the Matrix Tree Theorem that there are n^{2n-2} spanning trees on $K_{n,n}$ for all n.