Math 38. Graph Theory.

Solutions to Homework 5.

2.2.17. We will use the Matrix Tree Theorem to count the number of spanning trees of K_n. With the notation from the book, we have

$$Q = \begin{pmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & n-1 \end{pmatrix},$$

and so deletting the first now and column,

$$Q^* = \begin{pmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & n-1 \end{pmatrix},$$

where Q^* is an $(n-1) \times (n-1)$ matrix. By the Matrix Tree Theorem,

$$\tau(K_n) = (-1)^2 \det Q^* = \det \begin{pmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & n-1 \end{pmatrix} = \det \begin{pmatrix} 1 & -1 & \dots & -1 \\ 1 & n-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & -1 & \dots & n-1 \end{pmatrix},$$

where in the last step we added the sum of columns $2, 3, \dots, n-1$ to column 1, which does not change the determinant. Now, subtracting the first row from each of the other rows, we have

$$\tau(K_n) = \det \begin{pmatrix} 1 & -1 & -1 & \dots & -1 \\ 0 & n & 0 & \dots & 0 \\ 0 & 0 & n & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & n \end{pmatrix} = n^{n-2}.$$

2.3.3. The minimum cost is 21. Indeed, applying Kruskal's algorithm, the added edges are v_1v_2, v_2v_3, v_4v_5, v_2v_5 in this order, which have costs $3, 3, 7, 8$, respectively.

2.3.10. The proof that Prim's algorithm produces a minimum-weight spanning tree of G is very similar to the proof of Theorem 2.3.3, with the following difference.

Let T be the spanning tree produced by Prim's algorithm, and let T^* be a minimum-weight spanning tree that agrees with T in the longest possible initial sequence of edges. Let e be the first edge chosen from T that is not in T^*. When e is added to T^*, it creates a cycle C.

Let A be the set of vertices already reached by T before e is added, and let B be the remaining vertices. Then e connects a vertex in A with a vertex in B. So, the cycle C must have another edge e' connecting connecting a vertex in A with a vertex in B. Consider the spanning tree $T^* + e - e'$. When Prim's algorithm choses e, both e and e' connect a reached vertex with one not yet reached, so both edges are available. This means that the weight of e is less that or equal to the weight of e', and thus $T^* + e - e'$ is a minimum-weight spanning tree that agrees with T in a longer initial sequence of edges, in contradiction with the choice of T^*.

2.3.16. Let A, B, C, D the names of the people sorted from slowest to fastest. The four people can get across in 17 minutes as follows: CD cross (2 min), D goes back (1 min), AB cross (10 min), C goes back (2 min), CD cross (2 min).

Using graph theory, we define a bipartite graph G with

$$V(G) = \{u_S : S \subseteq \{A, B, C, D\}\} \cup \{v_T : T \subseteq \{A, B, C, D\}\},$$

where edges are pairs (u_S, v_T) where $T \subset S$ and $S \setminus T$ has 1 or 2 elements. Let the weight of such an edge be the time it takes for the slowest person in $S \setminus T$ to cross the bridge.

Think of u_S as indicating that the people S are on the original side of the bridge, and that the flashlight is on the same side. Think of v_T as saying that the people T are on the original side of the bridge, and that the flashlight is on the other side. Then, the edge (u_S, v_T) corresponds to the people in $S \setminus T$ crossing the bridge, and so the edges of G are the possible moves, the weight being the time they take.

The shortest path in G from $u_{\{A,B,C,D\}}$ (original state) to $v_\emptyset$ (final state, where everyone has crossed) describes the fastest way for the four people to cross the bridge, so we can use Dijkstra's algorithm to find it.

3.1.8. **True**. To prove it, notice that the components of the symmetric difference of two perfect matchings are even cycles, since every vertex has degree 0 or 2 in this symmetric difference. But since trees are acyclic, this implies that the symmetric difference has to be empty. Thus, a tree can't have two different perfect matchings.

3.1.24. One direction is straightforward: if a matrix is the sum of k permutation matrices, then its row and column sums equal k, since they are the sum of rows and column sums of the individual matrices.

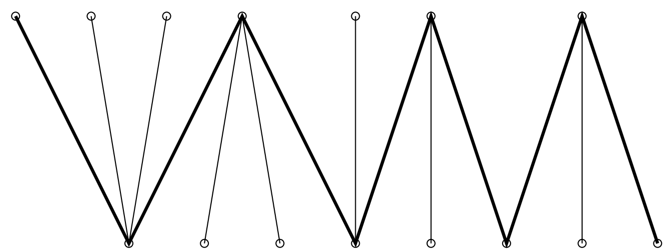
We prove the other direction (every matrix of nonnegative integers with row and column sums equal to k is the sum of k permutation matrices) by induction on k. Suppose the matrix is $n \times n$.

For $k = 1$ this is trivial, since there must be a 1 in each row and column.

Now we prove it for $k \geq 2$, assuming it is true for smaller values of k. Given a matrix A of nonnegative integers with row and column sums equal to k, construct a bipartite graph with vertices $\{u_1, \ldots, u_n\}$ and $\{v_1, \ldots, v_n\}$ so that the number of edges between u_i and v_j is the entry in row i and column j of A. This is a k-regular bipartite graph, so by Corollary 3.1.13, it has a perfect matching. The edges in the perfect matching define a permutation matrix, and subtracting it from A, we get a matrix of nonnegative integers with row and column sums equal to $k - 1$. By induction hypothesis, this matrix is the sum of $k - 1$ permutation matrices, so adding back the subtracted permutation matrix, we have expressed A as a sum of k permutation matrices.

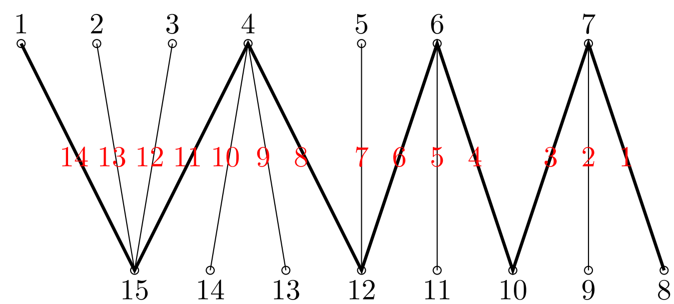
Bonus: Prove that every caterpillar has a graceful labeling.

A caterpillar is a tree, and thus bipartite. Consider a bipartition of its vertices into X and Y. Then the spine is a path $v_1 v_2 \dots v_k$ that alternates between X and Y. Suppose that $v_i \in X$ for odd i and that $v_i \in Y$ for even i. Draw the graph so that the vertices in X are above the vertices of Y, the spine goes across from left to write (alternating between X and Y), and for each v_i in the spine, its neighbors not in the spine are placed between v_{i-1} and v_{i+1} in the drawing. A picture is worth a thousand words:



The top vertices are in X, the bottom ones in Y, and the spine is drawn thicker in this picture.

Now assign to the vertices the numbers from 1 to n in clockwise order starting from the top left. In the example:



Then the differences between the endpoints of each edge are, from left to right, $n-1, n-2, \dots, 3, 2, 1$ (these are the red numbers in the picture). Therefore, this gives a graceful labeling for any caterpillar.