Math 725, Spring 2006
Solution Set #3

#1. [West 2.3.14] Let C be a cycle in a connected weighted graph. Let e be an edge of maximum weight in C; that is, $\mathrm{wt}(e) \geq \mathrm{wt}(e')$ for all $e' \in E(C)$. Prove that there is a minimum-weight spanning tree (MST) not containing e. Use this to prove that iteratively deleting a heaviest non-cut-edge until the graph is acyclic produces an MST.

Let T be an MST. If $e \notin T$ then there is nothing to prove, so suppose that $e \in T$. Let P be the path $C - e$, and let x, y be the endpoints of e. Then each of the two components H, H' of the forest $T - e$ contains one of the two vertices x, y. Let e' be an edge of P with one endpoint in each of H, H'; such an edge must exist because the endpoints of P are x and y. Then $T' = T - e + e'$ is a spanning tree that does not contain e, and moreover $\mathrm{wt}(e') \leq \mathrm{wt}(e)$, so $\mathrm{wt}(T') \leq \mathrm{wt}(T)$. Since T is an MST, equality must hold in the last expression, and in particular and T' is an MST that does not contain e.

Here is an explicit algorithm to compute an MST T of a connected weighted graph G:

```
$T_0$ := $E(G)$
$i$ := 0
while $T$ contains a cycle $C$ do
{
    $i$ := $i+1$
    Choose $e_i \in E(C)$ of maximum weight
    $T_i$ := $T_{i-1} - e_i$
}
$T$ := $T_i$
```

First, T_0 is connected, and e_i is not a cut-edge of T_{i-1} by Theorem 1.2.14, so by induction every T_i is connected. Moreover, the loop terminates only when T_i is acyclic. So the output T is a spanning tree.

Second, T_0 certainly contains an MST of G. By the first part of the problem, for each i, if T_{i-1} contains a spanning tree then so does T_i. Hence the output T contains, hence is, an MST. ■

- Many of you tried to prove the first part of the statement using Kruskal's algorithm. Unfortunately, doing so introduces more difficulties than it solves, since G may have several different MST's and you now need to figure out some clever order of the edges so that Kruskal's algorithm does not include e (particularly if C has more than one heaviest edge). Yes, you can do this, but why go through the hassle? When solving a problem like this, the right approach is usually to try to prove it by elementary methods (i.e., from the definitions of MST and the exchange rules), and then haul out the heavy machinery only if the simpler approach isn't sufficient.

#2. [West 2.2.1] Determine which spanning trees of K_n have Prüfer codes that (a) contain only one value; (b) contain exactly two values; or (c) contain $n-2$ distinct values.

Let T be a spanning tree of K_n, and let $p = (p_1, p_2, \ldots, p_{n-2})$ be its Prüfer code.

(a.) If $p_i = j$ for every i, then T is the star centered at j; that is, its edges are $\{jk : \; k \in [n] - \{j\}\}$. Equivalently, $T \cong K_{1,n-1}$.

(b.) If p contains exactly two values, then T has exactly two vertices that are not leaves. Such trees are called "double-stars" (see Exercise 2.1.46). By the way, the number of such trees is $\binom{n}{2}(2^{n-2} - 2)$.

(c.) If p contains all distinct values, then T has $n-2$ vertices of degree 2 and two leaves; that is, T is a path. (As we know, the number of paths in K_n is $n(n-1)\cdots(3) = n!/2$.) ■

#3. [West 2.2.16, modified] Let F_n be the "n-fan" defined by

$$\begin{aligned} V(F_n) &= \{v_0,\, v_1,\, \ldots,\, v_n\}, \\ E(F_n) &= \{v_1v_2,\, v_2v_3,\, \ldots,\, v_{n-1}v_n\} \cup \{v_0v_1,\, v_0v_2,\, \ldots,\, v_0v_n\}. \end{aligned}$$

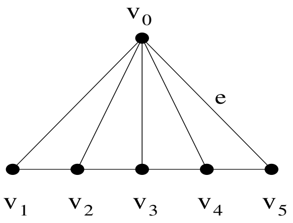
a. Prove that $\tau(F_n) = 3\tau(F_{n-1}) - \tau(F_{n-2})$.

b. Based on your argument in (a), describe a sequence of graphs $G_1, G_2, G_3, \ldots$ so that $a_i = \tau(G_i)$ is the i^{th} Fibonacci number; that is, $a_1 = a_2 = 1$ and $a_i = a_{i-1} + a_{i-2}$ for $i \geq 3$.

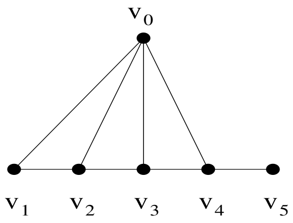
(a.) Let $e = v_0v_n \in E(F_n)$, and apply deletion-contraction:

$$\tau(F_n) \;=\; \tau(F_n - e) + \tau(F_n/e). \tag{1}$$

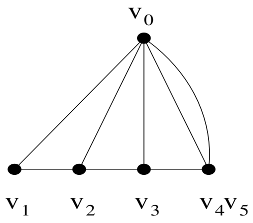
For example, F_5, $F - 5 - e$, and F_5/e are as follows:



F_5



$F_5 - e$



$F_5\ /\ e\ =\ H_4$

The graph $F_n - e$ looks like F_{n-1} with an extra leaf attached by a cut-edge. In particular, contracting that cut-edge recovers F_{n-1} and doesn't change the number of spanning trees: that is, $\tau(F_n - e) = \tau(F_{n-1})$.

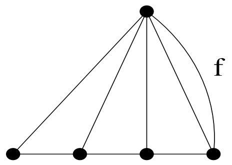
Meanwhile, $H_{n-1} := F_n/e$ looks like F_{n-1} with one additional edge f parallel to v_0v_{n-1}.

For simplicity of notation, set $f_n = \tau(F_n)$, $h_n = \tau(H_n)$. Then we can rewrite (1) as

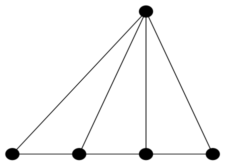
$$f_n \;=\; f_{n-1} + h_{n-1}. \tag{2}$$

Next, we apply deletion-contraction to H_{n-1}:

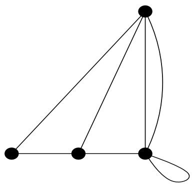
$$\tau(H_{n-1}) \;=\; h_n \;=\; \tau(H_{n-1} - f) + \tau(H_{n-1}/f). \tag{3}$$



$F_5\ /\ e\ =\ H_4$



$H_4 - f\ =\ F_4$



$H_4\ /\ f$

Now $H_{n-1} - f \cong F_{n-1}$, and H_{n-1}/f looks like H_{n-2} with an extra loop (which does not affect the number of spanning trees). So (3) becomes

$$h_{n-1} \;=\; f_{n-1} + h_{n-2}. \tag{4}$$

Since n was arbitrary, we can replace n with $n-1$ in (2) to obtain

$$f_{n-1} \;=\; f_{n-2} + h_{n-2}.$$

Solving for h_{n-2} gives

$$h_{n-2} = f_{n-1} - f_{n-2}.$$

Plugging that into (4) gives

$$h_{n-1} \;=\; 2f_{n-1} - f_{n-2}.$$

Then plugging that into (2) gives

$$f_n \;=\; 3f_{n-1} - f_{n-2}$$

as desired.

(b.) The desired sequence of graphs G_i is

$$H_0, F_1, H_1, F_2, H_2, F_3, H_3, \ldots$$

(where we set $H_0 := K_1$). Then $\tau(H_0) = \tau(F_1) = 1$, and the equations (2) and (4) comprise the Fibonacci recurrence. ■

#4. [West 2.2.18] Use the Matrix-Tree Theorem to compute $\tau(K_{r,s})$ for all numbers r, s.

Let the partite sets of $K_{r,s}$ be $X = \{x_1, \ldots, x_r\}$ and $Y = \{y_1, \ldots, y_s\}$. Note that $d(x_i) = s$ and $d(y_j) = r$ for all i, j. So the Laplacian matrix $L = L(K_{r,s})$ is

$$\left[\begin{array}{cccc|cccc}
s & 0 & \cdots 0 & -1 & -1 & \cdots & -1 & \\
0 & s & \cdots 0 & -1 & -1 & \cdots & -1 & \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots s & -1 & -1 & \cdots & -1 & \\
\hline
-1 & -1 & \cdots & -1 & r & 0 & \cdots 0 & \\
-1 & -1 & \cdots & -1 & 0 & r & \cdots 0 & \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
-1 & -1 & \cdots & -1 & 0 & 0 & \cdots r &
\end{array}\right].$$

Note that this is an $(r+s) \times (r+s)$ square matrix, with with r rows (resp. columns) in the top (resp. left) block and s rows (resp. columns) in the bottom (resp. right) block.

I'll exhibit a basis for $\mathbb{R}^{r+s}$ consisting of eigenvectors for L. Let η_i denote the i^{th} standard basis vector.

First, as usual, the "all-ones" vector

$$E_0 = \eta_1 + \eta_2 + \cdots + \eta_{r+s}$$

is a nullvector for L, that is, an eigenvector with eigenvalue 1.

Second, let

$$E_1 = (\eta_1 + \cdots + \eta_r) - (\eta_{r+1} + \cdots + \eta_{r+s});$$

we can calculate LE_1 directly (I omit the details) to see that it is an eigenvector, with eigenvalue $r + s$.

Third, for $2 \leq i \leq r$ we have

$$L(\eta_1 - \eta_i) = s(\eta_1 - \eta_i)$$

and for $r + 2 \leq j \leq r + s$ we have

$$L(\eta_{r+1} - \eta_j) = r(\eta_{r+1} - \eta_j).$$

Now, I claim that

$$\{E_0, E_1\} \cup \{\eta_1 - \eta_i : \; 2 \leq i \leq r\} \cup \{\eta_{r+1} - \eta_j : \; r + 2 \leq j \leq r + s\}$$

is a basis for $\mathbb{R}^{r+s}$. Indeed, it has cardinality $r+s$, and we have already seen (from looking at the Laplacian of K_n) that

$$\{(E_0+E_1)/2\} \cup \{\eta_1 - \eta_i :\ 2 \leq i \leq r\},$$
$$\{(E_0-E_1)/2\} \cup \{\eta_{r+1} - \eta_j :\ r+2 \leq j \leq r+s\}$$

are bases for the linear span of the first r and last s basis vectors, respectively.

The upshot is that the eigenvalues of $L = L(K_{r,s})$ are

$$0,\ r+s,\ \underbrace{r, r, \dots, r}_{s-1 \text{ times}},\ \underbrace{s, s, \dots, s}_{r-1 \text{ times}}.$$

Therefore, by the alternate version of the Matrix-Tree Theorem,

$$\tau(K_{r,s}) \ = \ \frac{(r+s)r^{s-1}s^{r-1}}{n(K_{r,s})} \ = \ r^{s-1}s^{r-1}. \quad \blacksquare$$

• It is also possible to apply row and column operations to the reduced Laplacian. However, you need to be clever about which reduced Laplacian; as one of you noticed, it is easier *not* to delete the same row and column. Instead, delete the r^{th} row and the $(r+1)^{st}$ column, to obtain a $(r+s-1) \times (r+s-1)$ square matrix $L' = L^{r,r+1}(K_{r,s})$ of the form

$$L' \ = \ \left[\begin{array}{c|c|c} S & 0 & -1 \\ \hline -1 & -1 & 0 \\ \hline -1 & -1 & R \end{array}\right]$$

with the row blocks of heights $r-1$, 1, $s-1$ from top to bottom, and the column blocks of the same widths left to right. Here S is a diagonal matrix with all diagonal entries equal to s, and R is a diagonal matrix with all diagonal entries equal to r.

Now, subtracting the r^{th} column from every column to its left gives the matrix

$$L'' \ = \ \left[\begin{array}{c|c|c} S & 0 & -1 \\ \hline 0 & -1 & 0 \\ \hline 0 & -1 & R \end{array}\right]$$

and now subtracting the r^{th} row from every row below it gives

$$L''' \ = \ \left[\begin{array}{c|c|c} S & 0 & -1 \\ \hline 0 & -1 & 0 \\ \hline 0 & 0 & R \end{array}\right].$$

This matrix is block upper-triangular, and these row and column operations haven't changed the determinant, so the Matrix-Tree Theorem gives

$$\tau(K_{r,s}) = (-1)^{r+(r+1)} \det L' = -\det L''' = s^{r-1}r^{s-1}.$$

#5. [West 2.1.58, modified] In this problem, you will prove the following theorem due to Smolenskii. Let S and T be trees with leaves $x_1, \dots, x_m$ and $y_1, \dots, y_m$ respectively, such that $\{d_S(x_i, x_j) = d_T(y_i, y_j)$ for all i, j. (Let's call the data $\{d_S(x_i, x_j) : \ 1 \le i < j \le m\}$ the *leaf-distance sequence* of T, or the LDS for short.) Then there is an isomorphism $f : S \to T$ such that $f(x_i) = y_i$ for every i.

a. Suppose that $f : G \to H$ is an isomorphism. Let $u \in V(G)$ and $v = f(u) \in V(H)$. Let G' be the graph constructed by attaching a leaf u' to G at u (that is, $V(G') = V(G) \sqcup \{u'\}$ and $E(G') = E(G) \sqcup \{uu'\}$) and let H' be the graph constructed by attaching a leaf v' to H at v. Show that $G' \cong H'$. (This is one of those statements that is intuitively true, but requires careful bookkeeping. In particular, you really have to use the definition of an isomorphism!)

Extend f to a bijection $V(G') \to V(H')$ by defining $f(u') = f(v')$. Then

$$f(N_{G'}(u')) = f(\{u\}) = \{v\} = N_{H'}(f(u'))$$

and

$$f(N_{G'}(u)) = f(N_G(u) \cup \{u'\}) = N_H(f(u)) \cup \{v'\} = N_{H'}(f(u))$$

and for $x \in V(G) - \{u, u'\}$,

$$f(N_{G'}(x)) = f(N_G(x)) = N_H(f(x)) = N_{H'}(f(x)),$$

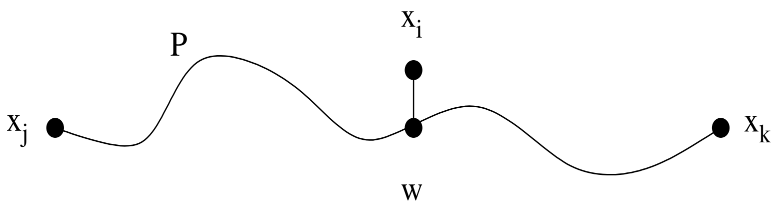
so we have verified that the extended function f gives an isomorphism $G' \cong H'$.

b. A *stub* of S is a leaf x_i whose unique neighbor in S has degree > 2. Fix a leaf $x_i \in V(S)$ and let w be its stem (unique neighbor). Call x_i a *stub* iff $d_T(w) > 2$. Show that x_i is a stub if and only if for some j, k,

$$d(x_i, x_j) + d(x_i, x_k) = d(x_j, x_k) + 2.$$

Conclude that x_i is a stub of S if and only if y_i is a stub of T.

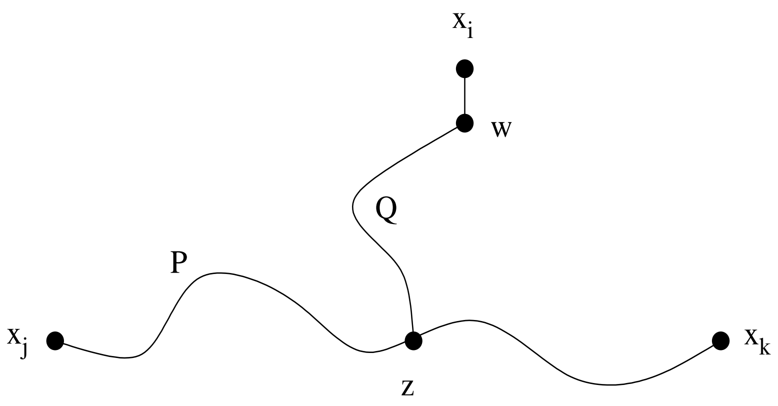
Suppose that x_i is a stub. Let $T' = T - x_i$, so that $d_{T'}(w) \ge 2$. Let P be a maximal path in T containing w but not v_i. Since w has at least two neighbors other than v_i, the path P will have as its endpoints two different leaves of T other than v_i; call these v_j and v_k. Let P_j and P_k be the v_j, w-subpath and w, v_k-subpath of P, respectively.



Then the unique x_i, x_j-path in T is $P_j \cup \{wx_j\}$, and the unique x_i, x_k-path in T is $P_k \cup \{wx_k\}$. In particular

$$d(x_i, x_j) + d(x_i, x_k) \ = \ (e(P_j) + 1) + (e(P_k) + 1) \ = \ e(P) + 2 \ = \ d(x_j, x_k) + 2.$$

Now suppose that there exist leaves x_j, x_k such that this equality holds. Let P be the path joining x_j and x_k, and let Q be a shortest path from x_i to a vertex $z \in P$. In particular Q contains no other vertex of P. Let P_j and P_k be respectively the v_j, z- and z, v_k-subpaths of P; note that each of these paths contains at least one edge (since z cannot be a leaf; it has a neighbor in P and a neighbor outside P). Then $Q \cup P_j$ and $Q \cup P_k$ are respectively the unique x_i, x_j- and x_i, x_k-paths in T.



Now

$$d(x_i, x_j) + d(x_i, x_k) = e(Q \cup P_j) + e(Q \cup P_k) = 2e(Q) + e(P)$$

and

$$d(x_j, x_k) + 2 = e(P) + 2.$$

By the desired equality, we have $e(Q) = 1$; that is, Q is the single edge $v_i w$. So $w = z \in P$, and w has at least two neighbors other than v_i, namely, one in each of P_j and P_k.

Finally, the property of being a stub is preserved under isomorphism, since if v_i is a leaf with stem w and f is an isomorphism, then $d(v_i) = d(f(v_i))$ and $d(w) = d(f(w))$. Therefore x_i is a stub if and only if y_i is.

c. Suppose that x_1 is not a stub. Describe the leaves and the LDS of $S - x_1$ in terms of those of S.

If x_1 is not a stub, then the leaves of $S - x_1$ are

$$\{w, x_2, x_3, \ldots, x_m\},$$

where w is the stem of x_1. The LDS of $S - x_1$ is given by

$$d_{S-x_1}(w, x_j) = d_S(x_1, x_j) - 1$$

for $2 \leq j \leq m$, and

$$d_{S-x_1}(x_j, x_k) = d_S(x_j, x_k)$$

for $2 \leq j < k \leq m$.

d. Suppose that x_i is a stub. Describe the leaves and the LDS of $S - x_i$ in terms of those of S.

If x_1 is a stub, then the leaves of $S - x_1$ are just $\{x_2, x_3, \ldots, x_m\}$, and $d_{S-x_1}(x_j, x_k) = d_S(x_j, x_k)$ for $2 \leq j < k \leq m$.

e. Prove Smolenskii's theorem by induction on the number of vertices. (Hint: Let x_1' and y_1' be the unique neighbors of x_1 and y_1 respectively. Show that there is an isomorphism $S - x_1 \to T - x_1$ mapping x_1' to y_1' and apply (a). This is easier in the case that the leaves x_1, y_1 are not stubs; if they are stubs, then you must use (b) to identify x_1' uniquely from the LDS of S.)

We would like to argue inductively as follows. Suppose that S, T are trees with the same order and LDS, with leaves $\{x_1, \ldots, x_m\}$ and $\{y_1, \ldots, y_m\}$ as above. Then the trees $S' = S - x_1$ and $T' = T - y_1$ have the same LDS (by (c) or (d)), so there is an isomorphism $f : S' \to T'$. However, it does *not* follow immediately from (a) that f can be extended to an isomorphism between S and T, since we do not know that f maps the stem of x_1 to the stem of y_1.

To take care of this subtlety, we will prove the following stronger statement by induction:

> Let S, T be trees of order n and leaves $\{x_1, \ldots, x_m\}$, $\{y_1, \ldots, y_m\}$ respectively, and suppose that $d_S(x_i, x_j) = d_T(y_i, y_j)$ for all i, j. Then there is an isomorphism $f : S \to T$ such that $f(x_i) = y_i$ for all i.

As a base case, this is trivial for trees with $n \leq 3$. Suppose that it holds for trees of order $< n$; we will prove it for trees S, T of order exactly n. Suppose that S, T have the same LDS, and let $S' = S - x_1$ and $T' = T - y_1$.

If x_1 is not a stub of S, then y_1 is not a stub of T, by (b) and the equality of LDS's. Let x_1' and y_1' be their respective stems. By (c), the trees S' and T' have the same LDS's, and by induction there is an isomorphism $f : S' \to T'$ such that $f(x_1') = y_1'$ and $f(x_i) = y_i$ for $2 \leq i \leq m$. By (a), defining $f(x_1) = y_1$ gives the desired isomorphism $f : S \to T$.

Now suppose that x_1 is a stub, so that y_1 is also a stub. By (b), again, S' and T' have the same LDS, and by induction, there is an isomorphism $f : S' \to T'$ such that $f(x_i) = y_i$ for $2 \leq i \leq m$. To extend f to an isomorphism $S \to T$, it suffices to show that $f(x_1') = y_1'$. Choose leaves x_j, x_k of S (hence of S') as in (b). Then the numbers

$$\begin{aligned} a &= d_S(x_j, x_1) - 1, \\ b &= d_S(x_k, x_1) - 1 \end{aligned}$$

sum to $d_{S'}(x_j, x_k)$, and so x_1' is uniquely determined as the vertex z on the (unique) x_j, x_k-path in S' such that

$$\begin{aligned} d_{S'}(x_j, z) &= a, \\ d_{S'}(x_k, z) &= b. \end{aligned}$$

On the other hand, f maps $z = x_1'$ to the unique vertex of T' at distance a from y_j and distance b from y_k, which is just y_1' (because S and T have the same LDS). To reiterate this, we have $f(x_1') = f(y_1')$, so f extends to the desired isomorphism $S \to T$. ■

Bonus problem: [West 2.3.16] Four people must cross a canyon at night on a fragile bridge. At most two people can be on the bridge at once, and there is only one flashlight (which can cross only by being carried). Kovalevskaia can cross the bridge in 10 minutes, Legendre in 5 minutes, Macaulay in 2 minutes, and Noether in 1 minute. If two people cross together, they move at the speed of the slower person. Oh, by the way, in 18 minutes a flash flood is going to roar down the canyon and wash away the bridge (together with anyone who isn't yet safe on the other side). Can the four people get across in time?

Draw a graph in which the vertices are labeled by subsets of $\{K, L, M, N, F\}$ (representing which people are still left on the unsafe side of the bridge; F stands for "flashlight"). The edges correspond to single crossings. Note that the graph is bipartite, since each edge must join a set containing F n to one not containing F.

Weight each edge by the time it takes to make a particular crossing; for example, the edge from $KLMF$ to K (representing Legendre and Macaulay crossing together to safety) has weight 5.

The quickest way to save all four people is represented by the shortest path from $KLMNF$ to $\emptyset$ in the weighted graph thus constructed. In fact, there is a path of weight 17:

$$KLMNF \xrightarrow{2} MNF \xrightarrow{1} M \xrightarrow{10} KLNF \xrightarrow{2} KL \xrightarrow{2} KLMNF$$

That is, proceed as follows:

(1) Send K and L across (2 minutes)
(2) Send K back (1 minute)
(3) Send M and N across (10 minutes)
(4) Send L back (2 minutes)
(5) Send K and L across (2 minutes)

Total: 17 minutes, and everyone's saved.