Math 724, Fall 2013
Homework #3 Solutions

(#1) Problem #82.

The inequality we want to prove is equivalent to the following statement: "In any complete graph G on $R(m-1,n)+R(m,n-1)$ or more vertices with edges colored red and blue, there is a red m-clique or a blue n-clique."

Fix a vertex x. Call y a "red neighbor" or "blue neighbor" of x according as the edge xy is colored red or blue.

The total number of neighbors of x is at least $R(m-1,n)+R(m,n-1)-1$. Therefore, either it has at least $R(m-1,n)$ red neighbors or it has at least $R(m,n-1)$ blue neighbors. (If neither of these cases are true, then the total number of neighbors of x is at most $(R(m-1,n)-1)+(R(m,n-1)-1) = R(m-1,n)+R(m,n-1)-2$, which is a contradiction.)

In the first case: x has at least $R(m-1,n)$ red neighbors. Let Y be the set of Call the set of red neighbors of x. By definition of $R(m-1,n)$, either Y has a blue n-clique and we're done — we have found a blue n-clique in G. Otherwise, Y has a red $(m-1)$-clique, when $Y \cup \{x\}$ is a red m-clique (remember, xy is a red edge for all $y \in Y$). blue neighbors Z.

The argument for the second case (x has at least $R(m,n-1)$ blue neighbors) is identical — switch "red" with "blue" and m with n.

(#2) Problem #83.

For the state of the art in Ramsey theory, consult S. Radziszkowski, "Small Ramsey Numbers," Electron. J. Combin. 1 (1994), Dynamic Survey 1, at `http://www.combinatorics.org/files/Surveys/ds1/sur.pdf`. According to this survey, the first calculation of $R(4,4)$ was in R.E. Greenwood and A.M. Gleason, "Combinatorial relations and chromatic graphs," Canad. J. Math. 7 (1955), 1–7. This article is accessible via the KU Libraries website, and is the source for the proof in (b).

(a) Problem #82 says that $R(4,4) \leq R(3,4)+R(4,3) = 2R(4,3)$ (since $R(p,q) = R(q,p)$ for all p,q). By Problem #81, we know that $R(4,3) \leq \binom{4+3-2}{4-1} = \binom{5}{3} = 10$.

(b) Here is the proof from the Greenwood–Gleason article cited above. I've added some details to their presentation.

Henceforth, all arithmetic will take place in the field $\mathbb{F}_{17} = \mathbb{Z}/17\mathbb{Z}$. Consider the graph G whose vertices are the elements of $\mathbb{F}_{17}$, with two vertices x,y joined by an edge iff $x-y \in \{\pm1,\pm2,\pm4,\pm8\}$. Observe that this is equivalent to the condition that $x-y$ is a nonzero *quadratic residue* (QR), i.e., a square modulo 17, because

$$\begin{array}{llll} 1^2 = 16^2 = 1, & 2^2 = 15^2 = 4, & 3^2 = 14^2 = -8, & 4^2 = 13^2 = -1, \\ 5^2 = 12^2 = 8, & 6^2 = 11^2 = 2, & 7^2 = 10^2 = -2, & 8^2 = 9^2 = -4. \end{array}$$

(A graph like this is called a *Paley graph.*) The construction is highly symmetric: for any $q \in \mathbb{F}_{17}$, the bijection $\phi : \mathbb{F}_{17} \to \mathbb{F}_{17}$ defined by $\phi(x) = x+q$ induces a graph isomorphism on G.

Suppose that $a,b,c,d \in \mathbb{F}_{17}$ form either a clique or a coclique in the Paley graph. By symmetry (subtract a from everything), we can assume WLOG that $a = 0$, and the nonzero numbers $b,c,d,b-c,b-d,c-d$ are either all QRs or all non-QRs. Multiplying everything in sight by b^{-1} gives

$$1, \quad C, \quad D, \quad 1-C, \quad 1-D, \quad C-D \tag{$*$}$$

(where $C = b^{-1}c, D = b^{-1}d$). The product of two QRs or of two non-QRs is a QR (this follows from the fact that the nonzero residues form a group under multiplication and the quadratic residues are a subgroup of index 2), so the six numbers in $(*)$ are distinct nonzero QRs. In particular, C and $1-C$ are distinct nonzero QRs, so the only possibility is $\{C, 1-C\} = \{2, -1\}$. But for the same reason $\{D, 1-D\} = \{2, -1\}$, and so the six numbers in $(*)$ cannot be distinct. This is a contradiction, and we conclude that G has neither a clique nor a coclique of size 4. It follows that $R(4,4) \geq 18$.

Here is Sage code to construct the Paley graph and check every 4-tuple of vertices by hand to see whether it is a clique or a coclique.

```
QRes = [1,2,4,8,9,13,15,16]
Edges = [[i,j] for i in range(17) for j in range(i+1,17) if (i-j) % 17 in QRes]
G = Graph(Edges)
Cliques = []
Cocliques = []
for X in subsets(range(17)):
    if len(X) == 4:
        EdgesX = G.subgraph(X).edges()
        if len(EdgesX) == 0:
            Cocliques.append(X)
        if len(EdgesX) == 6:
            Cliques.append(X)
Cliques,Cocliques
```

(c) In fact we can prove that $R(4,3) = 9$ (as opposed to 10). This will imply that $R(4,4) \leq 2R(4,3) = 18$, and together with (b) we will be able to conclude that $R(4,4) = 18$.

Suppose we have a 2-edge-colored K_9 that contains neither a red 3-clique or a blue 4-clique.

If any vertex x has six blue neighbors, then they contain either a red 3-clique (we're done) or a blue K_3 (which together with x forms a blue 4-clique).

If any vertex has 4 red neighbors, then they either have a red edge, i.e., a 2-clique (which together with x forms a red 3-clique) or a blue 4-clique (we're done).

So we can conclude that no vertex has more than five blue neighbors or more than 3 red neighbors. Therefore, EVERY vertex has exactly five blue neighbors and three red neighbors. But this contradicts Problem #69! The upshot is that the 2-edge-colored K_9 *must* contain either a red 3-clique or a blue 4-clique, i.e., $R(4,3) = 9$.

(#3) Problem #91.

The answer is $m_n = 2^n - 1$. We will prove this by induction.

The base case $n = 1$ clearly takes $2^1 - 1 = 1$ move. (We could even make $n = 0$ the base case — the solution takes $0 = 2^0 - 1$ moves.)

Call the rods A, B, C. Suppose that we know how to move a stack of $n-1$ discs from one rod to another in m_{n-1} moves. Then we can move a stack of n discs from rod A to rod C in $2m_{n-1} + 1$ moves as follows.

(1) Move the first $n-1$ discs from rod A to rod B. By induction, this can be done in m_{n-1} moves.
(2) Move the largest disc from A to C. This is a single move.
(3) Move the first $n-1$ discs from rod B to rod C. Again, this can be done in m_{n-1} moves

We have found a solution that takes $2m_{n-1}+1$ moves, so it follows that

$$m_n \leq 2m_{n-1}+1 \tag{0.1}$$

but we have not yet shown equality, because we need to rule out the possibility that there is some cleverer way to solve the problem of n discs.

On the other hand, suppose we have an optimal solution for n discs. At some point, the bottom disc must move from rod A to another rod, say C. Call this move M. The key observation is that M is a legal move *only if there is no other disc on either of these rods* — which means that every other disc is on rod B. Therefore, the sequence of moves before M must move the entire stack of $n-1$ discs from A to B, and the moves after M must move the entire stack of $n-1$ discs from B to C. Therfore, the entire solution must take at least $m_{n-1}+1+m_{n-1}=2m_{n-1}+1$ moves. We have proved that

$$m_n \geq 2m_{n-1}+1. \tag{0.2}$$

Combining (0.1) and (0.2), we see that $m_n = 2m_{n-1}+1$, and combining with the condition $m_0=0$, we deduce that $m_b = 2^n-1$.

(#4) Problem #116.

Let T be a labeled tree on n vertices with three vertices of degree 1. The Prüfer code of such a tree is a sequence $B=(b_1,\dots,b_{n-2})$ such that exactly three elements of $[n]$ (the leaves of T) are missing from the sequence; in other words, the set $\{b_1,\dots,b_{n-2}\}$ has cardinality $n-3$. For this to happen, there must be one number that occurs twice in B, and $n-4$ numbers that each appear once. In order to describe B, we can proceed as follows:

(1) Choose the three numbers to omit: $\binom{n}{3}$ possibilities.
(2) Choose which number x to repeat: $n-3$ possibilities.
(3) Choose which two locations get x's: $\binom{n-2}{2}$ possibilities.
(4) Assign the other $n-4$ numbers to the remaining locations: $(n-4)!$ possibilities.

So the final answer is

$$\boxed{\binom{n}{3}\binom{n-2}{2}(n-3)(n-4)!\ .}$$

There are many equivalent ways to do the counting and to write this formula, e.g.,

$$\binom{n}{3}\binom{n-2}{2}(n-3)!\ , \qquad \binom{n}{3}(n-3)\frac{(n-2)!}{2}\ .$$

For example, if $n=3$ then $\binom{1}{2}=0$, so the answer is 0, which makes sense — any three-vertex tree is a two-edge path, with one vertex of degree 2 and two of degree 1). If $n=4$ then the formula gives $\binom{4}{3}\binom{2}{2}\cdot 1! = 4$, which again is correct — the 4-vertex with three leaves are the stars, of which there are four (four ways to choose the center vertex).

Problem #5.

Every vertex of G must have degree a nonnegative integer in $\{0,1,2\dots,n-1\}$ (0 if it has no neighbors; $n-1$ if it has every other vertex as a neighbor). That is, there are n possibilities for the degree. Since there are n vertices, the only way that all vertices can have different degrees is if every degree occurs exactly once. In particular, there are vertices x,y such that $d_G(x)=0$ and $d_G(y)=n-1$. But $d_G(x)=0$ implies that xy is not an edge, and $d_G(y)=n-1$ implies that xy is an edge. This is a contradiction! So two vertices must have the same degree.

Note that the argument requires G to be simple (otherwise $d_G(y)=n-1$ would not force xy to be an edge.)

(Problem #6.)

$\boxed{(1) \iff (4)}$ Connectedness is by definition the condition that every two vertices of G are joined by at least one path. I claim that acyclicity is equivalent to the condition that every two vertices are joined by at *most* one path. Indeed, if G contains a cycle, then any two vertices on the cycle are joined by at least two paths (walking in different directions around the cycle. On the other hand, if

$$P = (v = a_1, a_2, \cdots, a_{n-1}, a_n = w), \qquad Q = (v = b_1, b_2, \cdots, b_{m-1}, b_m = w)$$

are two different paths joining v and w, then I can construct a cycle as follows. First, find the smallest i such that $a_i = b_i$ and $a_{i+1} \neq b_{i+1}$. (This is guaranteed to exist because $a_1 = b_1$ but $P \neq Q$.) Let

$$P' = (v' = a_i, a_{i+1}, \cdots, a_{n-1}, a_n = w), \qquad Q' = (v' = b_i, b_{i+1}, \cdots, b_{m-1}, b_m = w).$$

Now, let j be the smallest index $> i$ such that $a_j \in \{b_{i+1}, \cdots, b_{m-1}, b_m\}$. (Again, this is guaranteed to exist because $a_n = w$ belongs to that set.) Say $a_j = b_k$. Then

$$(v' = a_i, a_{i+1}, \cdots, a_{j-1}, a_j = b_k, b_{k-1}, \cdots b_{i+1}, b_i = v')$$

is a cycle, because by the construction of j, there are no repeated vertices other than $v' = a_i = b_i$. This proves the claim.

The key observation is that adding an edge $e = xy$ to a graph G does one of two things. If x, y were not in the same component, then adding e joins those two components into a single component, and therefore decreases the *number* of connected components by 1. Such an addition creates no new cycles — any *new* cycle would have to contain the new edge, but x, y are not linked by any path in G. On the other hand, if x, y are already in the same component, then adding e creates a cycle (specifically, if x, y were joined by a path P, then $P \cup e$ is a cycle in $G \cup e$) and does not change the number of components.

$\boxed{(1) \implies (2), (3)}$ Suppose G is connected and acyclic. Imagine building G by starting with n isolated vertices and adding one edge at a time. Since no cycle can ever be created, each edge addition must decrease the number of components by 1. But since we originally had n components (the isolated vertices) and wind up with 1 component (since G is connected), we must have added exactly $n - 1$ edges.

$\boxed{(2) \implies (1), (3)}$ Suppose G is connected and has $n - 1$ edges. As in the previous argument, imagine building G by starting with n isolated vertices and adding one edge at a time. Since we start with n components and wind up with 1 component, and the best each new edge can do is lower the number of components by 1 — and therefore the process never created a cycle.

$\boxed{(3) \implies (1), (2)}$ Same idea — start with n isolated vertices and add one at a time. No step created a cycle, so every step decreased the number of components by 1., There are $n - 1$ edges, so the resulting graph G has $n - (n - 1) = 1$ component, which says that G is connected.

(#7) (a) There are several proofs. The key observation is that every row of L sums to 0 (because each edge incident to vertex i contributes $+1$ to the diagonal entry in row i, but also contributes -1 to some other entry in that row). Likewise, every column sum is 0, because $L(G)$ is symmetric.

With this in hand, you can do an explicit calculation to show that L^i and L^j can be obtained from each other by row and column operations that end up preserving the determinant — this proof is elementary but arduous. You can also use a trick involving the cofactor matrix (also known as the adjugate); this is slick but less elementary. My favorite proof is the following, submitted by Kyle Maddox.

Let J_n be the $n \times n$ matrix whose entries are all 1. Consider $\det(M)$, where $M = L + J$. Every row and column of M sums to n.

Construct a matrix M' as follows: first replace the **first** column of M by the sum of all the columns; then, replace the **first** row by the sum of all the rows. So $\det(M') = \det(M)$ and

$$M' = \left[\begin{array}{c|ccc} n^2 & n & \cdots & n \\ \hline n & & & \\ \vdots & & L^1 + J_{n-1} & \\ n & & & \end{array}\right].$$

Now, multiply the top row of M' by $1/n$ to get M'', and then subtract the top row of M'' from each of the other rows to get M'''. That is,

$$M'' = \left[\begin{array}{c|ccc} n & 1 & \cdots & 1 \\ \hline n & & & \\ \vdots & & L^1 + J_{n-1} & \\ n & & & \end{array}\right], \qquad M''' = \left[\begin{array}{c|ccc} n & 1 & \cdots & 1 \\ \hline 0 & & & \\ \vdots & & L^1 & \\ 0 & & & \end{array}\right],$$

and $\det M''' = \det M'' = \frac{1}{n} \det M' = \frac{1}{n} \det M$. But evidently $\det M''' = n \det L^1$. So $\det L^1 = \frac{1}{n^2} \det M$. But the argument would work the same way if the "**first**"s in the construction of M' were changed to "jth", for any $j \in [n]$. It follows that $\det L^i$ is independent of the choice of i.

(b) Your examples should indicate that

$$\tau(G) = \begin{cases} 1 & \text{if } G \text{ is a tree,} \\ n & \text{if } G \text{ is a cycle of length } n, \\ n^{n-2} & \text{if } G = K_n. \end{cases}$$

Indeed, $\tau(G)$ is the number of spanning trees of G — this is the **Matrix-Tree Theorem.** Here is Sage code to do the calculations:

```
for n in range(1,8):
      Cn = graphs.CycleGraph(n)
      L = Cn.laplacian_matrix()
      L0 = L.submatrix(1,1)
      print(n, L0.det())                       # output: 1, 2, ..., 7
for n in range(1,8):
      Kn = graphs.CompleteGraph(n)
      L = Kn.laplacian_matrix()
      L0 = L.submatrix(1,1)
      print(n, L0.det())                       # output: 1, 1, 3, 16, 125, 6^4, 7^5
T1=Graph([(1,2),(1,3),(1,4)])                  # "star" tree on 4 vertices
T2=Graph([(1,2),(2,3),(3,4)])                  # "path" tree on 4 vertices
T1.laplacian_matrix().submatrix(1,1).det()     # output: 1
T2.laplacian_matrix().submatrix(1,1).det()     # output: 1
```