MIDTERM

MATH 38, SPRING 2012

SOLUTIONS

1. [20 points]

   (a) Prove that if G is a simple graph of order n such that $\Delta(G) + \delta(G) \geq n-1$, then G is connected. (Hint: Consider a vertex of maximum degree.)

   (b) Show that this bound is sharp (i.e. there is no smaller way to bound $\Delta(G) + \delta(G)$ below and guarantee G is connected).

Proof. For part (a), consider $v \in V(G)$ with $d(v) = \Delta(G)$. We can show that any vertex which is not a neighbor of v has at lease one neighbor in common with v as follows: Suppose $u \in V(G)$ is not adjacent to v, and consider $N(u) \cap N(v)$. Since $|N(u)| \geq \delta(G)$, $|N(v)| = \Delta(G)$, and $|N(u) \cup N(v)| \leq n-2$ (the size of $V(G) - \{u, v\}$), we have

$$\begin{aligned} |N(u) \cap N(v)| &= |N(u)| + |N(v)| - |N(u) \cup N(v)| \\ &\geq \delta(G) + \Delta(G) - (n-2) \\ &\geq n - 1 - (n-2) = 1. \end{aligned}$$

So all vertices in G have distance at most two from v, and thus G is connected.

For (b), consider $G = K_{n-2,1} + P_1$. Since $\Delta(G) + \delta(G) = n - 2 + 0$ and G is not connected, the bound in (a) is sharp. ☐

2. [20 points] Let $d_1 \geq d_2 \geq \cdots \geq d_n \geq 1$ be an integer sequence.

   (a) If this is a degree sequence for a forrest, calculate $S(n,k) = \sum_i d_i$ in terms of the number of vertices n and the number of components k.

   (b) With $S(n,k)$ as in part (a), show that every integer sequence $d_1 \geq d_2 \geq \cdots \geq d_n \geq 1$ with $\sum_i d_i = S(n,k)$ is the degree sequence for a forrest with k components.

(Hint: We already know the case when $k = 1$; you are welcome to cite that without proof.)

Proof. I wrote up a few solutions, for which these two lemmas come up a few times.

Lemma 1. *The following are equivalent:*

A. G is a forrest (acyclic) with k components.

B. G is acyclic with $n-k$ edges.

C. G has $n-k$ edges and k components.

D. G is loopless and every pair of vertices either has a unique path between them, or they are not connected.

The proof is by considering each component individually, and seeing that it is a tree.

Lemma 2. *If a strictly positive integer sequence of length n sums to $2n-2k$, then it must contain at least $2k$ 1's.*

We're now ready to proceed.

(a) By Lemma 1, G has $n-k$ edges and therefore its degrees sum to $2(n-k)$.

(b) The goal is, if you are given n integers satisfying

$$d_1 \geq d_2 \geq \cdots \geq d_n \geq 1 \qquad \text{and} \qquad \sum_{i=1}^{n} d_i = 2(n-k),$$

to find some forrest with $d(v_i) = d_i$ (for free, that forrest willl have k components by (a)). I have collected here several of your ideas for part b.

Method 1. Idea: Construction + Existance. Take out a clear-cut, and build a tree. Construct a forrest G as follows: Set $V(G) = \{v_1, \ldots, v_n\}$. Since $\sum_{i=1}^{n} d_i = 2(n-k)$ and $d_i \geq 1$, we have $d_{n-i} = 1$ for $i = 0, \ldots, 2k-1$ (the last $2k$ values, by Lemma 2). So connect v_{n-2i} to v_{n-2i+1} for $i = 0, \ldots, k-1$ to form $k-1$ P_2's. For the remaining $n-2k+2$ degrees, they sum to

$$\sum_{i=1}^{n-2k+2} d_i = 2(n-k) - 2(k-1) = 2((n-2k+2)-1)$$

and so there is some tree on $v_1, \ldots, v_{n-2k+2}$ which satisfies $d(v_i) = d_i$. Use these vertices to form such a tree, and the result is a forrest with $k-1+1 = k$ components.

Method 2. Idea: Construction. Use the Prüfer code.

We know $n \geq 2k$ since $d_i \geq 1$. Make a list of length $n-2k$ with values from $\{1, \ldots, n\}$ with $d_i - 1$ i's. Use the algorithm for trees, except now at the last step, there will be $2k$ unmarked vertices. Pair these up as you like, adding a single edge between pairs. The method guarantees that you have a forrest with $n-k$ edges, which therefore has k components. (With more thought, this should imply that there are no more than $\frac{(n-2k)!}{\prod_i (d_i-1)!} * \prod_{i=1}^{k-1} \binom{n-2i}{2}$ such trees [the last step of pairing them up is a little funny]).

Method 3. Idea: Existence + tinkering. There's some graph with this degree sequence, so manipulate any such graph until it is the desired forrest. Since $\sum_i d_i$ is even, there is some graph on n vertices with $d(v_i) = d_i$. Let G be one such graph. Since G has $n-k$ edges, G has at least k components. If all the components in G are acyclic, then G has exactly k components and G is a forrest (Lemma 1). Otherwise, find some component H which has a cycle, and let e be an edge in that cycle with endpoints u and v. Let e' be and edge in a different component H' with endpoints u' and v'. Then $G - e - e' + uu' + vv'$ is a graph with one fewer component, since e was not a cut edge in H, and everything in H' is now connected to H (either through u' or v'). Furthermore, this two-switch preserved the degrees of all vertices in G. By induction on the number of components, it is possible to arrive at a graph F with $d_F(v_i) = d_i$ which has k components (and is therefore acyclic and a forrest).

Method 4. Idea: Induction. Add a total of 2 somehow to the d_i's, and use induction to build a forrest with one too many edges and one too few components. Then figure out how to delete an edge and get back to the desired forrest.

Fix n. If $\sum_{i=1}^n d_i = 2(n-1)$, then we know there is some tree with $d(v_i) = d_i$. Otherwise, assume that $k > 1$ and that for any $k-1$ if

$$d'_1 \geq d'_2 \geq \cdots \geq d'_n \geq 1 \qquad \text{and} \qquad \sum_{i=1}^{n} d'_i = 2(n-(k-1)),$$

then there is a forrest on n vertices with $d(v_i) = d'_i$.

Try 1: Let $d'_1 = d_1 + 2$ and $d'_i = d_i$ for $i = 2, \ldots, n$. Then if $\sum_{i=1}^n d_i = 2(n-k)$, we have $\sum_{i=1}^n d'_i = 2(n-k) + 2 = 2(n-(k-1))$. So there is some forrest G with $k-1$ components with $d(v_i) = d'_i$. Since $d'_1 > 2$, we know v_1 has at least 2 neighbors, u and w. Delete the edges v_1u and v_1w, and add the edge uw. This has preserved the degree of u and w, it has dropped the degree of v_1 by 2, and it has preserved the acyclic nature of G (since it simply took the unique uw path and shortened it by one). So the result is a graph G' with $d(v_i) = d_i$, and which is acyclic, and is therefore a forrest with k components.

Try 2: Let $d'_1 = d_1 + 1$ and $d'_2 = d_2 + 1$. Again. by induction, there is some forrest with $k-1$ components with $d(v_i) = d'_i$.

<u>Case 1:</u> v_1 is adjacent to v_2 in G. Delete v_1v_2.

<u>Case 2:</u> v_1 is in the same component as v_2, but is not adjacent. Then consider the v_1, v_2 path in G. Let u_1 and u_2 be their respective neighbors on this path. Perform the following transformation:

- Delete v_1u_1 and v_2u_2. This has separated the component into three pieces, one with v_1, one with v_2, and one with neither.
- Splice in the component without v_1 or v_2 into anywhere else by picking any other edge ab (there is at least one since v_1 had at least two neighbors), deleting it and adding the edges au_1 and bu_2. This restores the degrees of u_1 and u_2, as well as the original number of components.

<u>Case 3:</u> v_i is in a different component than v_2. Let u_1 and u_2 be neighbors of v_1 and v_2 respectively (they exist because $d(v_i) \geq 1$). Exchanges the edges u_1v_1 and u_2v_2 for u_1u_2. This preserves the degrees of u_1 and u_2 and lowers the degrees of v_1 and v_2 by 1, as desired.

In all three cases, the result is a graph with one more component and one fewer edge, and so is a forrest with k components.

Method 5. Idea: Constuction + tinkering. Build a tree T on $n+1$ vertices so that $T - v_{n+1}$ is a forrest with the desired degree sequence.

Form a tree that corresponds to the following Prüfer code:

i. For $i = 1, \ldots, k-1$ alternate d_i occurrences of i with one occurrence of $n+1$;

ii. list $d_i - 1$ ocurrences of i for $i = k+1, \ldots, n$;

iii. list d_k ocurrences of k.

For example, if your integer sequence is

$$d_1 = 5, d_2 = d_3 = 3, d_4 = d_5 = d_6 = 2, \underbrace{d_7 = \cdots = d_{17} = 1}_{\text{11 ones}},$$

then $n = 17$, $k = 3$, and the resulting code is

$$\underbrace{1, 1, 1, 1, 1}_{d_1\ \text{1's}}, 18, \underbrace{2, 2, 2}_{d_2\ \text{2's}}, 18, 4, 5, 6, \underbrace{3, 3, 3}_{d_3\ \text{3's}}.$$

The result will be a list with $n-1$ values from the elements $\{1, \ldots, n+1\}$ and so will result in a tree T with $n+1$ vertices.

a. For $i = 1, \ldots, k$, since i appears d_i times, we have $d_T(v_i) = d_i + 1$;

b. for $i = k+1, \ldots, n$, since i appears $d_i - 1$, we have $d_T(v_i) = d_i$; and

c. since $n+1$ appears $k-1$ times, we have $d_T(v_{n+1}) = k$.

d. Since $n+1$ appears each time the each of $1, \ldots, k-1$ disappears from the list when building T, and so $n+1$ is adjacent to $v_1, \ldots, v_{k-1}$.

e. Once $n+1$ disappears from the list, it will not be used until no other values are available. This will not occur until the final step, when only v_k and v_{n+1} remain to be connected. So in T, v_{n+1} is adjacent, in total, to $v_1, \ldots, v_{k-1}$. (The best picture is going through the procedure of building the example above, and then observing the adjacencies of v_{18}).

Therefore, the induced graph $T - \{v_{n+1}\}$ is a forrest on $v_1, \ldots, v_n$ with k components with $d(v_i) = d_i$ (removing k edges from a T yields a forrest with $k+1$ components; removing the edges incident to v_{n+1} and then v_{n+1} itself gives the desired result).

Method 6. Idea: Construction. Build it explicitly, not relying on the Prüfer code.

We'll construct a forrest with $d(v_i) = d_i$ via a series of graphs with weighted vertices as follows. At each step, every vertex in the graph G has a non-negative integer weight. At the end, the weights will all be 0 and G will the the desired forrest.

Step 1: *Place the $2k$ vertices $v_n - 2k + 1, \ldots, v_n$ in G, and weight them with "1".*

Justification: We know there are at least $2k$ ones in the integer sequence, so this can be done.

Step 2: *Add $v_1, v_2, \ldots, v_{n-2k}$ sequentially as follows:*

When adding v_i, look for a vertex v_j already in G (so $j < i$ or $j > n - 2k$) with the largest nonzero weight. Add the edge $v_i v_j$, and update the weight of v_j by subtracting 1. Weight v_i with $d_i - 1$. When breaking ties, add v_i to a component with the fewest vertices with nonzero weight.

Justification: In order to be able to add the next vertex, you only require that some vertex in G has weight at least 1, which is equivalent to showing that the sum of the weights is positive. The weight is a working tally of

$$\text{wt}(v_i) = \#(\text{edges } v_i \text{ needs}) - \#(\text{edges } v_i \text{ has}) = d_i - d_G(v).$$

Furthermore, at each step G is a forrest with $2k$ components. So by (a),

$$\sum_{v\in V(G)} d_G(v) = 2(2k+i-2k) = 2i.$$

Therefore we require that, for $i = 1, \ldots, n-2k$,

$$\begin{aligned} 0 \leq \sum_{v_i\in V(G)} \mathrm{wt}(v_i) &= \sum_{v_i\in V(G)} (d_i - d_G(v_i)) \\ &= \left(2k + \sum_{j=1}^{i} d_i\right) - \left(\sum_{v\in V(G)} d_G(v_i)\right) \\ &= 2k + \left(\sum_{j=1}^{i} d_j\right) - 2i \\ &= 2k + \sum_{j=1}^{i}(d_j - 2). \end{aligned}$$

But $\sum_{j=1}^{i}(d_j-2)$ is a non-decreasing function in i while $d_i \neq 1$ and then is a decreasing function (so the global minimum happens at an endpoint). When $i = 1$, $\mathrm{wt}(v_1)$ is at least 1 (since $d_1 \geq 1$ and $k \geq 1$); when $i = n-2k$,

$$\begin{aligned} 2k + \sum_{j=1}^{n-2k}(d_j - 2) &= 2k + \left(\sum_{i=1}^{n} d_i\right) - 2k - 2(n-2k) \\ &= 2(n-k) - 2(n-2k) \\ &= 2k. \end{aligned}$$

So we can always add the next vertex.

Step 3: *When all of the vertices have been added, there will be one vertex with weight 1 in each of the $2k$ components. Pair up the components and add an edge between weight 1 vertices in each pair.*

Justification: Revisiting the above calculation, we showed that at the moment when the last vertex is added,

$$\sum_{v\in V(G)} \mathrm{wt}(v) = 2k.$$

There can be no more than $2k$ vertices with nonzero weight since weights are non-negative. There also can be no fewer: We always added a new vertex to an old vertex of highest available weight, and (aside from the first $2k$ vertices) we added vertices in decreasing order of initial weight. Therefore, the only way for one component to have no non-zero weight vertices is for all vertices in G to be of weight at most 1. Moreover, by how we break ties, if some component is "closed", no other component can have more than one vertex of weight 1 (or we would have closed a vertex in that component first). So the $2k$ weights must be spread evenly across the components.

At the end, every vertex has weight 0 (and so $d(v_i) = d_i$). So you have a graph with $n-k$ edges, and exactly k components, so G is a forrest.

□

3. [15 points] Recall that a *tournament* is a directed complete graph, i.e. for every two vertices u and v, either there is an edge from u to v or there is an edge from v to u (but not both). Show that every tournament has a spanning path. (Hint: Can a non-spanning path be exchanged for a longer path?)

Proof. Consider a maximal non-spanning path $P = p_1 \to p_2 \to \cdots \to p_\ell$ in a tournament G. Then for any $u \notin V(P)$, we have $p_1 \to u$ and $u \to p_\ell$ (because P was maximal). So there exists some pair p_i and p_{i+1} satisfying $p_i \to u \to p_{i+1}$, and hence

$$p_1 \to \cdots \to p_i \to u \to p_{i+1} \to \dots p_\ell$$

is a path of length greater than P. Therefore, any maximum path in G is spanning. □

4. [15 points] Show that a m-regular simple graph G has a decomposition into copies of $K_{1,m}$ if and only if G is bipartite.

Proof. **If G is bipartite**, with partites X and Y, then since G is m-regular,

$$\text{for each } v \in X, \qquad G[\{v\} \cup N(v)] \cong K_{1,m}.$$

Every edge in G appears as an edge incident to some $v \in X$ and every $u \in Y$ appears in a neighborhood of some v (since $d(u) = m > 0$), and no edge appears incident to more than one $v \in X$ (because G is bipartite. So G decomposes as $\bigcup_{v \in X} G[\{v\} \cup N(v)] \cong \bigcup_{v \in X} K_{1,m}$.

If G decomposes into copies of $K_{1,m}$, then let X be the set of vertices occurring as a center of some star, and let Y be the set of vertices occurring as leaves of some star. The decomposition implies that $X \cup Y = V(G)$. A priori, these two sets might not be disjoint, but since G is m-regular, no vertex in G can have more than m neighbors. Therefore, no vertex in X can also be a vertex of Y (or vice versa), so $X \cap Y =$. Again, by the decomposition, two vertices are only adjacent if one is in X and one is in Y (they are adjacent in some star), and so X, Y forms a bipartition of G. □

5. [10 points] Give an example of a weighted graph G with $n(G) = 4$ where...
   (a) ...executing Dijkstra's algorithm from any vertex will give a minimum weight spanning tree.
   (b) ...executing Dijkstra's algorithm from any vertex will not give a minimum weight spanning tree (no matter where you start it, you will not get a minimum-weight spanning tree).
   (c) ...every minimum weight spanning tree has the property that the distance between any two vertices in G is the same as the distance between those two vertices in the tree.
   (d) ...every minimum weight spanning tree has the property that the distance between some two vertices in G is strictly less than the distance between those two vertices in the tree (i.e. all shortest paths are lost between some pair of vertices).

Answer. For (a) and (c), pick any weighted tree on four vertices (there is exactly one spanning tree, and it is of minimum weight, and contains all paths present in G). For (b), take the K_4, drawn as a square with two diagonals. Weight the two horizontal edges by 2, the two verticals by 4, and the two diagonals by 5. The minimum weight spanning tree has weight $2+2+4 = 8$, but Dijkstra's algorithm will give a tree of weight $2+4+5 = 11$ from any vertex. For (d), let $G = C_4$ with equal weights on all edges. Any spanning tree is a path P, and the endpoints of the path will be farther apart in P than they are in G.

6. [10 points] List 8 or more (up to 20) facts about the Petersen graph.
(Show me your diversity of knowledge about graph theory thus far. What kinds of questions does one ask about a graph? In addition to giving properties that the Petersen graph has, it's also legitimate to list properties that the Petersen graph doesn't have, or to count things having to do with the Petersen graph. Eight legitimately diverse facts, including a couple of non-trivial statements, will receive full credit; if in doubt, list more or ask.)

1. It has 10 vertices (size 2 subsets of [5]) and 15 edges ($E(G) = \{uv \mid |u \cap v| = 0\}$).
2. It's three-regular.
3. It has girth 5 (no cycles of length 1-4), and various other facts about cycles of length longer.
4. It decomposes in various interesting ways, but not into claws, and not into edge-disjoint spanning trees.
5. It's not bipartite.
6. Its chromatic number is 3.
7. It's simple.
8. There are 5! isomorphic graphs on [10] to the Petersen graph.
9. It's connected, and admits an orientation which yields a strongly connected digraph.
10. It has a spanning path, and some spanning caterpillars, and some large number of spanning trees.
11. It has no Eulerian path or circuit (too many odd vertices).
12. The shortest non-extendable trail has length 8 (since it has to contain a cycle, and then a walk between vertices in that cycle).
13. It represents one isomophism class with its degree sequence, but not the only such class. In fact, the graph generated by the algorithm for graphic sequences generates a different one (one of the bipartite graphs) and the algorithm for simple graphs generates a third (5 P_2's with loops at each vertex).
14. It has several perfect matchings. Therefore, any closed walk will have to duplicate at least five edges.
15. It's not planar.
16. It is the union of 5-cycles, and so has no cut-edges and no cut-vertices.
17. It has radius and diameter 2, and therefore is its own center.
18. Its complement also has diameter and radius 2.
19. Every pair of non-adjacent vertices share a unique neighbor.
20. Someone more ambitious than I could write down the adjacency and incidence matrices.

7. [10 points] For free: tell me what you like most about graph theory so far (a favorite topic, kind of problem, way of thinking about things, etc.) or something you've learned related to graph theory (from class or not) that you enjoy a lot.

   Clearly, it's you guys that makes this class great for me!!