Problem Set 2 (due Wednesday, September 24)

Instructions:

- The assignment is due at the beginning of class on the due date specified. Late assignments will not be accepted.
- We encourage you to attempt and work out all of the problems on your own. You are permitted to study with friends and discuss the problems; however, *you must write up you own solutions, in your own words.*
- If you do collaborate with any of the other students on any problem, please list all your collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is prohibited.
- We require that all homework submissions be neat, organized, and *typeset*. You may use plain text or a word processor like Microsoft Word or LaTeX for your submissions.

1. (10 points) Hopping from one minimum spanning tree to another

Let G be a weighted connected undirected graph.

(a) Prove that if T and T' are two minimum spanning trees of G, then there exists a sequence $\langle T_0, \ldots, T_k \rangle$, $k \geq 0$, such that: (i) T_i is a minimum spanning tree of G, $0 \leq i \leq k$, (ii) $T_0 = T$, (iii) $T_k = T'$, and (iv) $|T_{i+1} \setminus T_i| = 1$, $0 \leq i < k$ (i.e., T_i and T_{i+1} differ in exactly one edge).

(b) Prove that if T and T' are two minumum spanning trees, then T and T' have the same "weight distribution" (i.e., for any weight w, both T and T' contain the same number of edges with weight w).

2. ($5 + 3 \times 5 = 20$ points) Broadcast via gossiping

The paradigm of gossiping is being considered as a robust mechanism for spreading information in a distributed network, or influence in a social network. Suppose we have an undirected connected network G with n nodes. A node, say r, has a piece of information M that it wants to broadcast to the entire network. Consider the following gossiping protocol.

In each step, each node that has a copy of M, sends a copy of M to a neighbor chosen uniformly at random. Assume that all the nodes are sychronized in their steps.

In this exercise, we want to place a bound on the *completion time* of the above protocol; that is the number of steps it takes before every node receives a copy of M.

(a) Write programs to simulate the above protocol on the following graphs: the star graph (one node having an edge to $n-1$ others), the complete graph (n nodes with all pairwise edges), the line, and the $\sqrt{n}\sqrt{n}$ two-dimensional grid. Plot the expected completion time, for each graph, as a function of n.

The rest of this exercise presents a sequence of steps leading to an analysis.

(b) Suppose a node u has a copy of M and degree d. What is the expected number of steps, in terms of d, before u sends a copy of M to a specific neighbor v?

(c) Let P be a shortest path from u to v. Prove that the sum of the degrees of all the nodes on P is at most $3n$.

(d) Using parts (b) and (c), derive an upper bound, in terms of n, on the expected number of steps it takes for an arbitrary node v to receive a copy of M.

Unfortunately, part (d) does not give us a bound on the expected completion time, since it only bounds the time taken for an arbitrary node v – not *all nodes* – to receive M.

(e) Let us revisit part (b). Again, suppose a node u has a copy of M and degree d. Find an upper bound, in terms of d, on the number of steps it takes for a specific neighbor v of u to receive a copy of M from u with probability at least $1 - 1/n^3$.

(f) Using parts (c) and (e), derive an upper bound, in terms of n, on the number of steps it takes for an arbitrary node v to receive a copy of M with probability at least $1 - 1/n^2$. Argue that the same bound yields an upper bound on the number of steps it takes for *all nodes* to receive a copy of M with probability at least $1 - 1/n$.

3. (10 points) Clustering to maximize separation

Many scientific applications, including medical imaging, classification of astronomical objects, and web document categorization, require partitioning a given set of objects into disjoint sets. Here is one of many variants of clustering problems. You are given a set S of n points $v_1, \ldots, v_n$, together with their pairwise distances. You may assume that the distances are symmetric and nonnegative, and that the distance between a point and itself is zero, while that between two distinct points is nonzero. Let $d(u, v)$ denote the distance between two points u and v. For any two nonempty sets X and Y of points, define the *distance* $d(X, Y)$ *between* X *and* Y to be

$$\min_{u \in X, v \in Y} d(u, v).$$

Given a number $m \leq n$, you are asked to partition S into m nonempty disjoint sets $S_1, \ldots, S_m$ so as to maximize the minimum distance between any pair of disjoint sets; that is, determine disjoint sets S_1 through S_m that maximizes $\min_{i \neq j} d(S_i, S_j)$ subject to the constraint that $\cup_i S_i = S$.

Give an algorithm to solve the above problem. Analyze the running time. Make your algorithm as efficient as you can, in terms of its worst case running time.

4. (10 points) Communication on Planet Anti-Huffman

Alice and Bob find themselves in the strange planet of *Anti-Huffman*. Suspicious of their neighbors as usual, they decide to use an encoding scheme for communication. Their encoding is based on a set S of m *code words* that they both share.

Alice wants to send a data string D of length n to Bob. Alice would like to determine whether D can be encoded as a concatenation of a sequence of code words from S. Furthermore, if the encoding exists, she would like to determine an encoding that uses the minimum length sequence of code words.

For example, consider the binary alphabet and let $S = \{0, 10, 0101\}$, and $D = 0101010100$. The encoding that splits D as $0101; 0; 10; 10; 0$ has length five, while a minimum-length encoding that splits D as $0101; 0101; 0; 0$ has length four. On the other hand, there is no encoding for the string 111 using code words in S.

Alice would like to solve the encoding problem efficiently. Fortunately for her, planet Anti-Huffman only supports *prefix-full codes*. A set S of code words is prefix-full if it satisfies the following property: if $s \in S$, then every nonempty prefix of s is also in S. An example of such a set is $\{0, 1, 01, 011, 010, 0111, 01110, 01111\}$. (A string x is a *prefix* of string y if there exists another string z such that y is a concatenation of x and z; that is, $y = x; z$. Thus, for example, 011 is a prefix of 01110 since $01110 = 011; 10$.)

Give an efficient algorithm for Alice to determine a minimum-length encoding of a given string D using code words from a prefix-full code. If no such encoding exists, then the algorithm must indicate so. Justify the correctness of your algorithm. Analyze the running time of your algorithm. Make your algorithm as efficient as you can, in terms of its worst-case running time.