Problem Set 2 (due Wednesday, October 6)

1. (5 points) Approximate median via sampling

Sampling is a common technique for solving many estimation problems. Given a set A of n numbers, and $\varepsilon \geq 0$, define an ε-approximate median of A to be an element whose rank is between $\lfloor (1-\varepsilon)n/2 \rfloor$ and $\lfloor (1+\varepsilon)n/2 \rfloor$.

Suppose we would like to find an ε-approximate median of A very quickly, even when n is very large. A natural strategy is to take a set S of c elements, chosen uniformly at random from A, and then return the median of S.

How large should c be, in terms of ε, so that the probability that the answer returned is an ε-approximate median is at least $1 - \varepsilon$?

2. (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).

3. (15 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) Optional: Run experiments on special graphs such as the star graph, the complete graph, the line, the grid, etc., to estimate the expected completion time, 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 $1 - 1/n$.

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.