Practice Midterm 2 (answers on the back)

On the midterm, you may use any algorithm described in class without giving its pseudocode or re-deriving its running time.

1. Indicate whether the following statements are True or False. You do not need to give a proof.

   ______ If $G = (V, E)$ is a connected, undirected graph then the minimum spanning tree of G contains the $|V| - 1$ edges whose weights are smallest.

   ______ Kruskal's minimum spanning tree algorithm does not always produce a minimum spanning tree if the connected, undirected graph has negative weight edges.

   ______ Any correct algorithm to determine if an array of $n \geq 2$ bits contains a 0 followed (immediately) by a 1 must examine every bit in the worst case.

   ______ Every connected, undirected graph G contains a vertex whose removal from G keeps G connected. (Removing a vertex from a graph implies removing all its adjacent edges.)

   ______ The expected number of times I need to toss a coin that comes up "heads" with probability 1/3 until I get a "head" is 3.

2. Let $G = (V, E)$ be an undirected weighted graph, and let F be a subgraph of G that is a forest (i.e. F does not contain any cycles). Design an efficient algorithm to find a spanning tree in G that contains all the edges of F, and has minimum cost among all spanning trees containing F.

3. Suppose we want to broadcast a message from a particular machine A to all the machines it can reach. We want to minimize the time it takes for the message to reach each of these machines. We know, for every link (A, B), the delay $d(A, B)$ caused by sending a message from A to B across this link. (If no such link exists, the delay is infinite.) We also know, for every machine B, the delay $t(B)$ in re-transmitting a message through that machine. Thus, if the message travels from A to B and then from B to both C and D, it takes time $t(A) + d(A, B) + t(B) + d(B, C)$ to reach C and $t(A) + d(A, B) + t(B) + d(B, D)$ to reach D. Describe an efficient algorithm that finds the optimal forwarding strategy for each machine in order to minimize the time for a message from machine A to reach each other (reachable) machine. A forwarding strategy specifies, for each machine, the machines to re-transmit an incoming message to. The total number of re-transmissions should be $n - 1$.

4. There are n houses on the north side of Main Street. The city wants to determine where to place a fire station on the vacant south side of Main Street so that the sum of the distances from each house to the fire station (as measured along Main Street) is minimized.

   Let x_i be the location of the ith house on Main Street. The city wants to find a value y such that

   $$\sum_{i=1}^{n} |x_i - y|$$

   is minimized.

   What value of y works? Prove that it does. Describe an efficient algorithm that finds this value of y.

Solution Sketches

1. False.

   False.

   False.

   True.

   True.

2. Run Kruskal's algorithm on G, but start with F as the initial set of edges. In other words, insert the edges of F into the minimum spanning tree first.

3. Create a directed graph G with two vertices X_{in} and X_{out} for each machine X. The edges of the graph G are (1) $(X_{\mathsf{out}}, Y_{\mathsf{in}})$ with weight $d(X,Y)$ for each link (X,Y); and (2) $(X_{\mathsf{in}}, X_{\mathsf{out}})$ with weight $t(X)$ for each machine X.

   The length of a path in G from A_{in} to X_{in} is the delay to send a message from machine A to machine X. Run Dijkstra's single source shortest paths algorithm to find the shortest paths from A_{in} to all other vertices in G. The children of X_{out} in this tree are the machines that X should re-transmit messages to. The total number of retransmissions is $n-1$. The running time is $O((n+m)\log n)$ where n is the number of machines and m is the number of links.

4. The median of $x_1, \ldots, x_n$ works.

   For convenience in the proof, reorder the x_i so that $x_1 \le x_2 \le \cdots \le x_n$. Consider the contribution $|x_i - y| + |x_{n-i+1} - y|$ to the total sum for each pair of houses x_i, x_{n-i+1} for $1 \le i \le \lfloor n/2 \rfloor$. The smallest contribution, $(x_{n-i+1} - x_i)$, is obtained if y lies between x_i and x_{n-i+1}. Thus

$$\sum_{i=1}^{n} |x_i - y| \ge \sum_{i=1}^{\lfloor n/2 \rfloor} x_{n-i+1} - x_i.$$

   Choosing y to be the median of $x_1, \ldots, x_n$ achieves this lower bound.

   The running time to find y is $O(n)$ using linear time k-Select.