ECE 454/750T10, Spring 2014 — Assignment 3
Due Fri, June 13, 11:59:59 PM

(Your submission must be typeset, and in pdf. Submit to the dropbox on Learn. If you are in 454, mention the names of both group members. Only one of the group members should make a submission.)

1.(5 points) Suppose we are given a weighted undirected connected graph $G = \langle V, E \rangle$ and $S \subseteq V$. Disprove the following assertion. The smallest subtree of an MST of G that contains all the vertices from S is a Steiner tree of S in G.

2.(10 points) Suppose T is a tree. Show that $|E[T]| = |V[T]| - 1$.

3.(7 points) In the proof for the correctness of Prim's algorithm in the lecture notes, it says, "Assume that we have an MST, T, with edge $\langle x, y \rangle$ that is on the unique path $u \rightsquigarrow v$ in T, and crosses the same cut of the graph."

Show that in every MST T of G, there is exactly one such edge $\langle x, y \rangle$ that exists.

4.(15 points) Assume that the only operations on a file are read and write. Assume also that we are dealing with the "remote access model" — the canonical file is on the server only.

There are two kinds of delays that are incurred when a remote client makes a read or write request to the server: (i) network delays, and, (ii) delays at the client and server. Assume that all instances of delays are nondeterministic, even if we seek to read or write the same file.

(a) The UNIX notion of correctness is, "a read(*file*) returns the effect of the last write(*file*). And, if we have two write(*file*) operations that overlap in time, the last write takes effect."

Suppose we adopt no protection mechanisms whatsoever at the client or server, and allow any client that requests to read or write a file at any time to take effect. Show a scenario in which the UNIX notion of correctness is not met.

(b) Suppose we clarify the notion of UNIX correctness to be, "...the last write to complete takes effect." Now can we assert, without any protection mechanisms at a client or server, that the UNIX notion of correctness is met?

(c) Suppose, in addition to read(*file*) and write(*file*), a client can also issue start transaction and end transaction messages to the server. All read and write requests between a start transaction and end transaction either all take effect, or none takes effect. A client is allowed to have at most one transaction going at any given time.

What should the server do when it receives start transaction and end transaction to ensure the UNIX notion of correctness? Your solution should ensure that no client starves. That is, if a client wants the server to process its read and/or write requests, the server should eventually satisfy such a client's requests, even if that client has to retry its attempt, perhaps several times.

5.(5 points) Suppose we have two machines in a distributed system, each with a clock that is supposed to tick 1000 times per millisecond. One of them actually does, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?

6.(5 points) To achieve totally-ordered multicasting with Lamport timestamps, is it necessary that every message is acknowledged?