Problems due as noted.

Assignment

Problems identified by x.y(z) denote the problem “y”, in chapter “x” of the textbook, with part “z”. If “z” is not noted, then the entire problem is required.

Due Mar 23

- 5.15 (a,b,c) If not possible, explain why.
- 5.18 (a,b) Remember smallest is left. When there are choices, put most recent node on the left.
- 5.31 Think greedy. This is very similar to Huffman encoding.
- 5.9 (a) Prove = proof, disprove = counter-example

Due Mar 26

- 5.2 (a) Prim only, track set S and cost and prev arrays, as show in class example.
- 5.5 (a,b) For each statement either give a proof of correctness, or a counter example.
- 5.10 Show that = prove = give a good argument for the truth of the statement
- 5.9 (d) Prove = proof, disprove = counter-example

Due Mar 28

- 5.2 (b) Kruskal only (track order of edges processed, members of X, and the disjoint sets for each step.)
- 5.12 You are giving a sequence of operations that insure the height of the tree is maximized.
- 5.9 (f) Prove = proof, disprove = counter-example

Due Mar 30

- 5.28 Discover a greedy algorithm
- 5.32 Discover a greedy algorithm
- 5.9 (h) Prove = proof, disprove = counter-example

Due Mar 30

- Write a program to find the weights of the MSTs of as many graphs as possible that are available from the `/usr/citlocal/cs3510/graphs` on the CIT computer systems. Record the average runtime for each size. Submit the MST weights and runtime chart with your assignment. Chart relevant theoretical limits. The graph filenames have the format `[graph-n-s.txt]` where `n` is the number of vertices and `s` is a graph number. For example, `[graph-20000-2.txt]` is the second graph with 20000 vertices. Each file has the number of vertices, followed by edge descriptions. The vertices are numbered 1 - `n`.

Submission

- For the written work, at the beginning of class, on the due dates, submit paper copies of your solutions.
- For the experimental determination, at the beginning of class on the due date, submit paper copies of the graphs.