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.

Assignment 5a, Due Mar 7

• 5.9 (a) Prove = proof, disprove = counter-example
• 5.2 (a) Prim only, track set S and cost and prev arrays, as shown in class example.

Assignment 5b, Due Mar 9

• 5.5 (a,b) For each statement either give a proof of correctness, or a counter example.
• 5.12 You are giving a sequence of operations that insure the height of the tree is maximized.
• (5 points) Implement Prim’s algorithm for MST’s. You may need to modify your heap code to work with it. Or, you may be creative enough to use it as is. Sample input files are available at /usr/citlocal/cs3510/graphs on the department computers. Bring a printout of the code to class. Also, bring a list of MST weights for at least 3 of the smaller graphs. 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.

Assignment 5c, Due Mar 19

• 5.2 (b) Kruskal only (track order of edges processed, members of X, and the disjoint sets for each step.)
• 5.10 Show that = prove = give a good argument for the truth of the statement
• (5 points) Implement a wrapper, timing program for your Prim’s algorithm code. Bring a printout of the code to class.

Assignment 5d, Due Mar 21

• 5.9 (d) Prove = proof, disprove = counter-example
• 5.9 (f) Prove = proof, disprove = counter-example
• (5 points) Record the average run time for Prim’s algorithm on each size of graph. Also, record the weight of the MST’s found. Bring a printout of the table of times and weights to class.

Assignment 5e, Due Mar 23

• 5.15 (a,b,c) If not possible, explain why.
• 5.32 Discover a greedy algorithm
• 5.9 (h) Prove = proof, disprove = counter-example

Assignment 5f, Due Mar 26

• 5.18 (a,b) Remember smallest is left. When there are choices, put most recent node on the left.
• 5.28 Discover a greedy algorithm
• (5 points) Chart the measured run time of Prim’s algorithm, along with the usual set of theoretical run time curves, properly normalized. Bring a printout of your chart.

Assignment 5z, Due Never

• 5.31 Think greedy. This is very similar to Huffman encoding.

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.