CS 3510: Algorithms

Assignment 5

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

- 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

- 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.
- 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.
  Find 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 vetices, followed by edge descriptions. The vertices are numbered 1 - `n`. Submit these MST weights in a table.

Assignment 5c

- 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
- Implement a wrapper timing program for your Prim’s algorithm code.

Assignment 5d

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

Assignment 5e

- 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

- 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
- 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

- Submit you solutions by the due date and time. For written problems, your work and answers as a PDF to Canvas. For code, submit the source code to the class git repository. For tables and graphs, submit a PDF to Canvas.