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.

TO BE UPDATED BELOW HERE

Due Mar 7 Shortest Paths

• 4.1 Run Dijkstra, tracking the problem data in a table.
• 4.14 By efficient, we mean no worse than Dijkstra’s algorithm.
• 4.22 Read this problem, and write down your questions about how to turn it into a graph path problem, such that our path algorithms can be modified to solve it.

Due Mar 9 Shortest Paths

• 4.2 Run Bellman-Ford, tracking the problem data in a table as we did in class. Each iteration is a new array, based on the previous array. Start from node S.
• 4.11 How can you find cycles using path algorithms in this chapter?
• 4.19 Look for a modified version of Dijkstra that meets the criteria.

Due Mar 19 Shortest Paths

• 4.8 Prove = proof, disprove = counter-example
• 4.12 Your algorithm should be O(|V|^2) or better.

Due Mar 21 Shortest Paths

• 4.5
• 4.15

Due Mar 21 Heaps (15 points)

• Implement the binary heap from Figure 4.16. Measure the performance of decreasekey(), deletemin() and makeheap() as a function of the number of elements in the heap. Create a chart with the runtimes of each of these functions plotted. Include theoretical curves to compare with.

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.