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 1 Dijkstra
- 4.1 Run Dijkstra, tracking the problem data as we did in class.
- 4.14 By efficient, we mean no worse than Dijkstra's algorithm.
- 4.22 Read this problem, and write down your questions about it. Don’t solve it.

Due Mar 1 NOT REQUIRED
- 4.19 Look for a modified version of Dijkstra that meets the criteria.
- 4.22 (a) This is a mathematical question, using the formulas in the problem. A negative cycle is a path that starts and ends at the same location, and has a total path cost less than 0.

Due Mar 3 Heaps
- 4.8 Prove = proof, disprove = counter-example
- 4.12 Your algorithm should be O(|V|^2) or better.
- 4.22 (b) Similar to part (a).

Due Mar 15 Bellman-Ford
- 4.2 Run Bellman-Ford, tracking the problem data as we did in class.
- 4.11 How can you find cycles using algorithms in this chapter?
- 4.15 Assume you have a solution to problem 4.5. How can you quickly solve this problem?
- 4.22 (c) Your algorithm should leverage the work from parts (a) and (b). Your running time analysis may use \( r^* \) instead of \( R \).
- 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 graph with the runtimes of each of these functions plotted.

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.