

Problem Set 8

Due: Thursday, December 13

This is an *optional* problem set. Any points you receive on it will be counted as extra credit points. You are expected to know the material covered on this problem set for the final exam. Unlike other extra credit work, which can be handed in until Sat. Dec. 22, any work done on this assignment must be handed in by 5pm Thu. Dec. 13, so that I can post the solutions soon after than time.

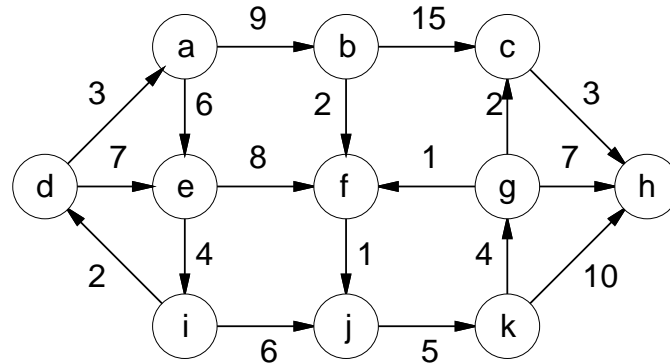
Reading: Handouts #36–37; *CLRS* Section B.4–B.5, Chapters 22 & 24; *CLR* Section 5.4–5, Chapters 23 & 25.

Suggested Problems: *CLRS* 22.1-1, 22.3-11, 22.4-2, 22.5-1, 22.5-6, 22-1, 22-2, 24.3-1, 22.3-3, 22.3-6, 22.3-7; *CLR* 23.1-1, 23.3-9; 23.4-2, ; 23.5-1, 23.5-3, 23.5-4, 23.5-6; 23-1, 23-2, 25.2-1, 25.2-3, 25.2-5, 25.2-6

Required Problems: You should write up and turn in the solutions to the problems listed below: The points awarded per problem are given in brackets.

Problem 1 [45]

Consider the following weighted directed graph G :



In the following problems, you should assume that the elements of `vertices[G]` are ordered alphabetically, and that the vertices within each adjacency list are also ordered alphabetically.

- a. [15] Draw the tree the is induced by performing a depth-first-search starting at node d . Label each vertex by its discovery and finish times. Using dotted lines, draw the edges of G that are not in the depth-first tree and label them as forward edges (F), back edges (B), or cross edges (C).
- b. [5] Consider the directed acyclic graph (DAG) G' that is the subgraph of G that results from removing from G all edges labeled as back edges (B) in part (a). Give a topological sort of G' .
- c. [5] Use the strongly connected component algorithm discussed in class (and found in *CLRS* 22.5/*CLR* 23.5) to determine the strongly connected components of G .

- d. [5] Draw the tree that is induced by performing breadth-first search on G starting at node d .
- e. [15] Use Dijkstra's algorithm to determine the shortest paths from d to each of the other vertices. Draw the shortest path tree that is induced by running Dijkstra's algorithm, and annotate each vertex by its shortest path distance from d .

Problem 2 [10] Do *CLRS* Exercise 22.2-5/*CLR* Exercise 23.2-5.

Problem 3 [20] (*CLR* Exercise 23.2-6; I cannot find a corresponding exercise in *CLRS*.) Give an efficient algorithm to determine if a given graph is bipartite. See *CLRS/CLR* for a definition of a bipartite graph.

Problem 4 [10] (This is a modified version of *CLRS* 24.3-2/*CLR* 25.2-2.) Dijkstra's algorithm assumes that the edge weights of the given graph are non-negative. Here we consider some of the consequences of allowing negative edge weights.

- a. [5] A negative-weight cycle of a weighted graph is a cyclic path whose path weight is negative. Suppose that G is a directed graph with a negative-weight cycle. Explain why the single-source shortest path problem may not be well-defined on such a graph.
- b. [5] If a graph G has some edges with negative weights but no negative-weight cycles, then the single-source shortest path problem is still well-defined. However, Dijkstra's algorithm is not guaranteed to correctly solve the problem in the presence of negative edge weights. Construct a connected, directed, weighted graph with four vertices such that Dijkstra's algorithm gives an incorrect solution to the single-source shortest path problem.

Problem 5 [15]

- a. [10] Do *CLRS* 22.4-3/*CLR* 23.4-3
- b. [5] Does your algorithm from part (a) work for *directed* graphs? Explain.

Extra Credit 1 [20] Do *CLRS* 22.4-5/*CLR* 23.4-5.

Problem Set Header Page
Please make this the first page of your hardcopy submission.

CS231 Problem Set 8

Due Thursday, December 13

Name:

Date & Time Submitted:

Collaborators (*anyone you worked with on the problem set*):

*In the **Time** column, please estimate the time you spend on the parts of this problem set. Please try to be as accurate as possible; this information will help me design future problem sets. I will fill out the **Score** column when grading your problem set.*

Part	Time	Score
General Reading		
Problem 1 [45]		
Problem 2 [10]		
Problem 3 [20]		
Problem 4 [10]		
Problem 5 [15]		
Extra Credit [20]		
Total		