Illinois Institute of Technology
Department of Computer Science

Third Examination

CS 430 Introduction to Algorithms
Fall, 2014

Wednesday, December 3, 2014
10am–11:15pm, 121 Life Sciences
11:25am–12:40pm, 113 Stuart Building

Print your name and student ID, *neatly* in the space provided below; print your name at the upper right corner of *every* page. Please print legibly.

Name:	
Student ID:	

This is an *open book* exam. You are permitted to use the textbook, any class handouts, anything posted on the web page, any of your own assignments, and anything in your own handwriting. Foreign students may use a dictionary. *Nothing else is permitted*: No calculators, laptops, cell phones, Ipods, Ipads, etc.!

Do all five problems in this booklet. *All problems are equally weighted, so do not spend too much time on any one question.*

Show your work! You will not get partial credit if the grader cannot figure out how you arrived at your answer.

Question	Points	Score	Grader
1	20		
2	20		
3	20		
4	20		
5	20		
Total	100		

1. **Union-Find**

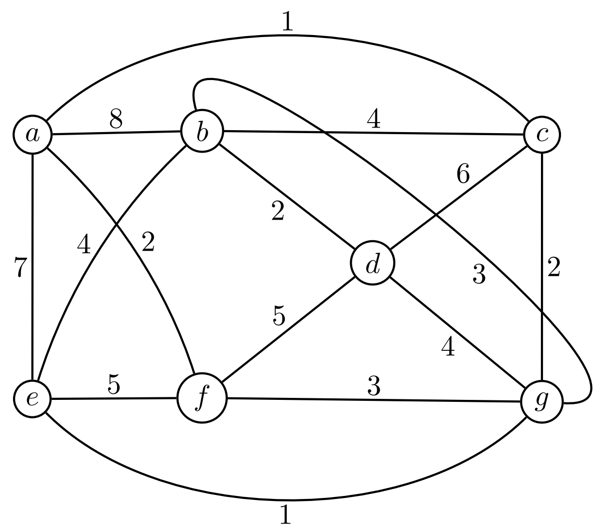
   We have jobs $1, 2, \ldots, n$ to be scheduled on on an arbitrarily large cluster of processors. There are m "same processor" constraints $(s_1, t_1), (s_2, t_2), \ldots, (s_m, t_m)$ meaning that jobs s_i and t_i must be scheduled on the same processor. There are also k "different processor" constraints $(u_1, v_1), (u_2, v_2), \ldots, (u_m, v_m)$ meaning that jobs u_i and v_i must be scheduled on different processors.

   (a) Give an algorithm using the union-find data structure of section 21.3 of CLRS3 to determine whether it is possible to assign jobs to processors so that all constraints are satisfied.

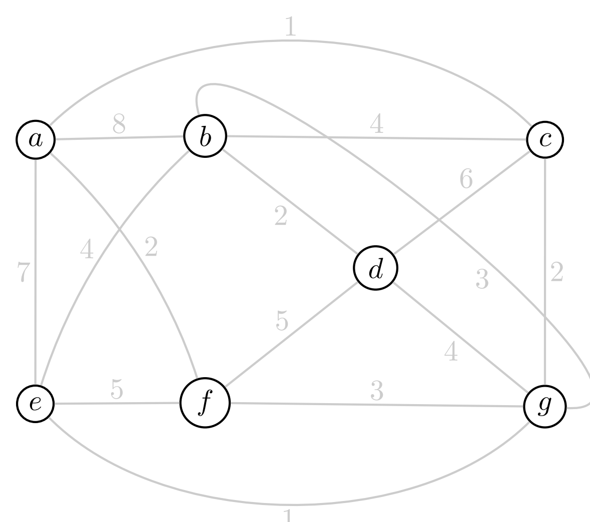
   (b) Analyze the total running time of your algorithm using the amortized analysis of Theorem 21.14 on page 581 of CLRS3.

2. **Basic Graph Algorithms**

   Consider the following graph G:

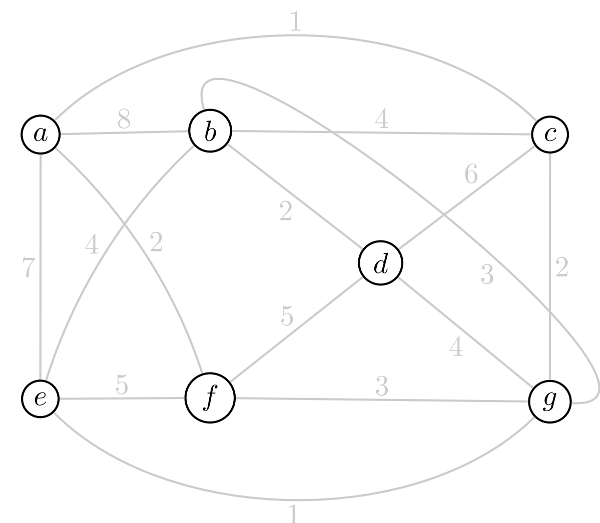


   (a) Show a breadth-first search tree of G, including the discovery time $v.d$ and the predecessor vertex $v.\pi$ for each vertex v. Do not show the edges that BFS ignores.

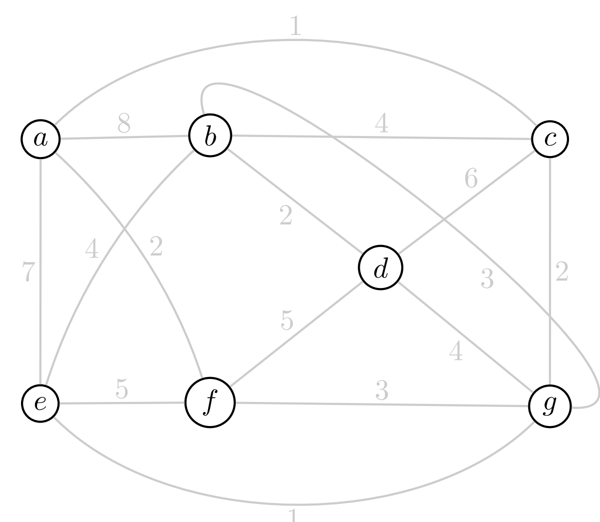


2. **Basic Graph Algorithms, continued.**

   (b) Show a depth-first search tree of G, including the back edges, the discovery time $v.d$, the finishing time $v.f$, and the predecessor vertex $v.\pi$ for each vertex v.

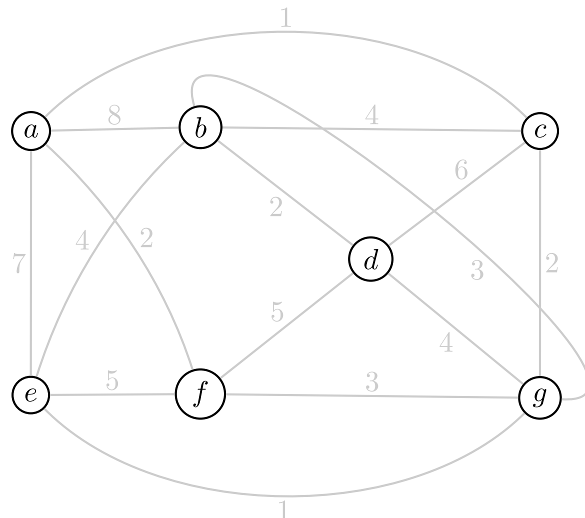


   (c) Show a minimum spanning tree generated by Kruskal's algorithm.



2. **Basic Graph Algorithms, continued.**

(d) Show a minimum spanning tree generated by Prim's algorithm with starting vertex (root) a.



(e) Explain the differences/similarities between the trees in parts (c) and (d).

3. **Shortest Paths**

   Consider a connected, weighted, directed graph G. Define the *thickness* of a path to be the maximum weight of any edge on the path.

   (a) Modify Dijkstra's algorithm to determine the path of minimum thickness between two vertices of G.

   (b) Does it affect your algorithm if the edge weights may be negative?

   (c) Prove your modification is correct.

   (*Hint*: This is a very easy problem.)

4. **Polynomial-Time Reductions**

   Consider the two closely related problems:

   - HAM-PATH: Given an undirected graph G, determine whether G contains a simple *path* that visits every vertex exactly once.
   - HAM-CYCLE: Given an undirected graph G, determine whether G contains a simple *cycle* that visits every vertex exactly once.

   Describe polynomial-time reductions from HAM-PATH to HAM-CYCLE and from HAM-CYCLE to HAM-PATH.

   (*Hint*: You can add/delete edges/vertices to G and use, say, an algorithm for HAM-CYCLE a polynomial number of times to determine HAM-PATH.)

5. **SUBSET-SUM Reduction**

   In the proof of Theorem 34.15 (pages 1097–1100 in CLRS3), the target value used, for example, is 1114444 in Figure 34.19. Explain why a target values 1113333, 1112222, 1111111 are not adequate to prove the theorem.