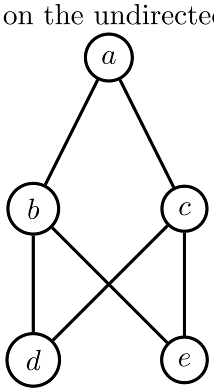
CMPS 101 Written homework 5, Spring 2014

5 problems, 10 points, due *at the start of class* Monday, June 2.

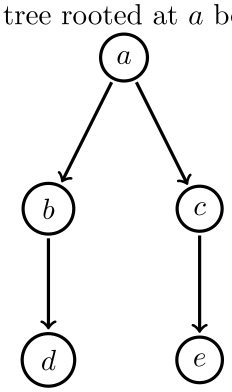
Students may submit this homework as individuals are in pairs. *Pairs must rotate*, so if you pair up with someone on this assignment, you may not pair up with that student on any previous or future written or programming assignment. Both members of a pair must fully understand their pair's solution, and pairs should submit *one* set of solutions with both students' names on the first page. The student's name and ucsc id (`@ucsc.edu` email address) should be in the upper right corner and multiple sheets should be stapled together in the upper left corner. Solutions to the problems should be in order and clearly labeled with the problem number. Although no points are given for neatness, illegible and/or poorly organized solutions can be penalized at the grader's option. See the course policies on acknowledging sources and protocol for study groups.

1. (2 pts) If Breadth-First search is started at any vertex a in any (undirected) graph G it will find a shortest path tree rooted at a. By changing the orders of the adjacency lists, Breadth-First search can often find a different shortest path tree from source a.

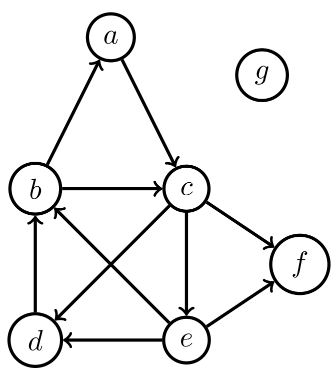
   Explain why BFS when run with source a on the undirected graph:

   will *never* find the shortest path tree rooted at a below.

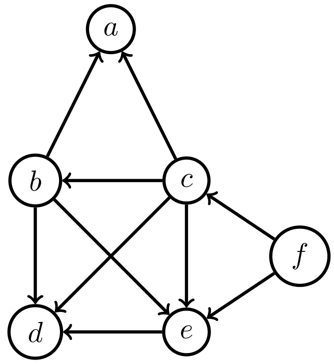
2. (2 pts) Perform a depth-first-search on the following graph G. Assume that the main DFS loop considers vertices in alphabetic order, and that the adjacency lists are also in alphabetic order.

   Draw the resulting DFS forest and list the forward edges, backward edges, and cross edges.

3. (2 pts) Give an example directed graph G where some vertex v has both leaving and entering edges, but yet v is the only vertex in its DFS tree. Assume that the vertices are lettered, the "for each vertex" loop of DFS examines vertices in alphabetical order, and the adjacency lists are in alphabetical order. Illustrate the DFS by drawing the resulting DFS forest.

4. (1 pt) List the vertices in the following graph in (one of the) topological orders.



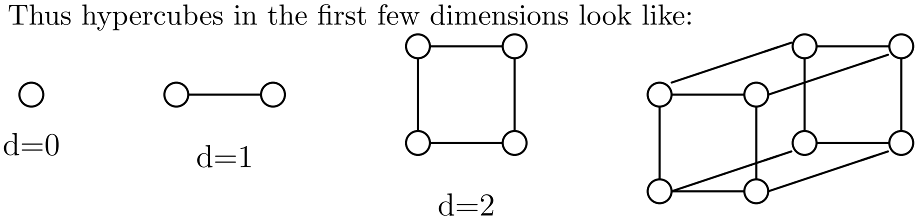
5. (3 pts) Run Dijkstra's algorithm on the directed graph of Figure 24.2 using vertex z as a the start vertex (*not* vertex s). Show the d and π values *after each iteration of the while loop* in the style of Figure 24.6.

Recommended induction problem (not to be turned in)

A hypercube is special kind of graph. Hypercubes are defined recursively as follows: A d-dimensional hypercube is the undirected graph constructed as follows:

> The 0-dimensional hypercube is just a single vertex (with no edges). For $d > 0$, the d-dimensional hypercube consists of two copies of the $(d-1)$-dimensional hypercube with additional edges crossing between the copies to join the corresponding vertices.

Thus hypercubes in the first few dimensions look like:



d=0 d=1 d=2 d=3

Use the fact that a d-dimensional hypercube has exactly 2^d vertices to prove formally by induction that that d-dimensional hypercubes have have exactly $d2^{d-1}$ edges. Carefully adhere to our style of induction proof (clearly identifying your Inductive hypotheses and what is assumed and what is to be shown in each step). and carefully define any notation that you use.

(Hint: for $d > 0$, the number of edges in a d-dimensional hypercube is 2^{d-1} crossing edges, plus twice the number of edges in a $d-1$ dimensional hypercube.)