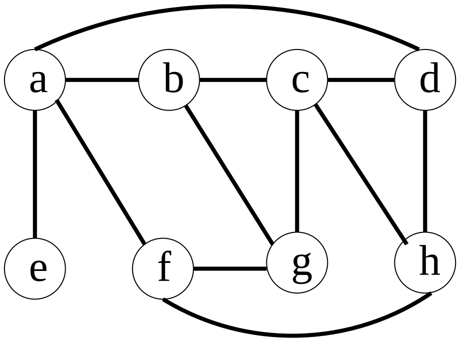
CS 3343 Analysis of Algorithms – Fall 2010

11/2/10

9. Homework

Due **11/16/10** before class



1. **Adjacency matrix (3 points)**
   Give the adjacency matrix representation for the graph above. Assume that vertices are ordered alphabetically.

2. **Adjacency lists (3 points)**
   Give the adjacency lists representation for the graph above. Assume that vertices (e.g., in an adjacency list) are ordered alphabetically.

3. **Depth-first search (6 points)**
   Consider a depth-first traversal of the graph above, starting at vertex a. Assume the graph is given in your adjacency lists representation of question 2. Mark the results of the following subquestions in a copy of the drawn graph.

   (a) (2 points) Give the discover time (d-value) and the finish time (f-value) of each vertex.

   (b) (2 points) Draw the depth-first tree.

   (c) (2 points) Mark each edge with its DFS classification (tree edge, back edge, forward edge, cross edge)

4. **Breadth-first search (4 points)**
   Consider a breadth-first traversal of the graph above, starting at vertex a. Assume the graph is given in your adjacency lists representation of question 2. Mark the results of the following subquestions in a copy of the drawn graph.

   (a) (2 points) Give the visit time stamp for each vertex (according to the pseudo code on slide 7).

   (b) (2 points) Draw the breadth-first tree.

Flip over to back page ⟹

5. **Adjacency lists vs. adjacency matrix (7 points)**

(a) (3 points) Give pseudo-code to convert a graph given in adjacency lists representation to its adjacency matrix representation. What is the runtime?

(b) (4 points) Both DFS and BFS include the following for loop referring to vertices `v` and `w`:

```
for each w adjacent to v do{
   // some statement
}
```

Give pseudo-code that implements this loop using (i) adjacency lists and (ii) an adjacency matrix. Analyze the runtime for both (assume that the statement inside the loop takes O(1) time).