CSCI 241, Fall 2012

Assignment #5: Graphs (30pts + 15 bonus points) Name _____

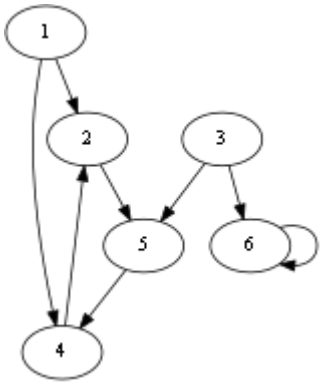
Summary

- **Goal**: This assignment is worth 10% of the grade for the course. In this assignment you will be playing with graphs. Note that questions like these are fair game on the final exam, so you may want to play with more examples on your own.
- **Collaboration policy for this homework**: For this homework, you may not collaborate with another student, although you may work together to understand the course material in a general way.
- Due time: **9 AM in class, Friday, Dec 7.**
- Hand in: A well-stapled-together printout of this handout (double-sided if at all possible), with your name and answers written clearly on it.

Tasks

1. (12 pts) The *transpose* of a directed graph G = (V, E) is a graph $G^T = (V, E^T)$, where $E^T = \{(v, u)$ in V x V: (u, v) in E}. Thus G^T is G with all its edges reversed. For the following graph G

   a. (4pts) Give both the adjacency-list and adjacency-matrix representation.



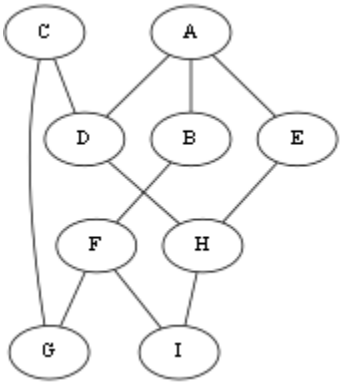
   b. (4pts) Give both the adjacency-list and adjacency-matrix representation for G^T.

   c. (2pts) Can the following code snippet produce G^T for the given G?

```
for i = 1 to n  //n: # of vertices of G
  for j = i+ 1 to n
      if A[i][j] == 1 then  // A is the adjacency matrix for G, B for G^T
         B[j][i] = 1;
```

   d. (2pts) Have some discussion on which representation you would prefer in computing G^T from G.

2. (12pts) Using the given undirected graph G,



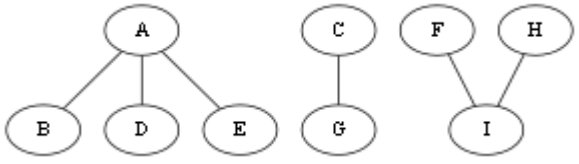
   a. (4pts) Complete the following trace table for the depth-first traversal algorithm. Whenever a new vertex to process must be selected and there is more than one possibility, use the vertex with the smallest number.

step	Node processed	Stack contents	"reached" nodes	"unreached" node set
0		A	A	B,C,D,E,F,G,H,I
1	A	BDE	A,B,D,E	C,F,G,H,I
2				
		…		

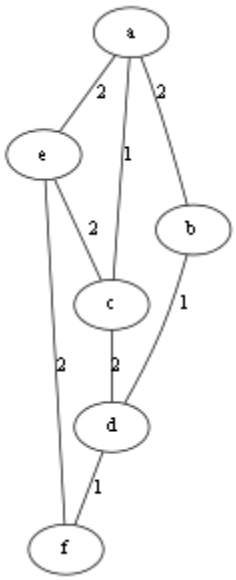
   b. (1.5pt) Write down the output from the above steps (in order of nodes being processed).

   c. (2.5pts) draw a corresponding subgraph from the output in (b). Is this subgraph a connected graph?

   d. (4pts) If G is not a connected graph as follows, how to modify the above process to get all connected components for G?



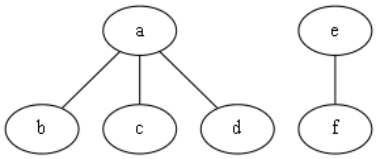
3. (6pts) For the following weighted graph G,
   a. (3pts) Give a minimal spanning tree



b. (3pts) Is this minimal spanning tree for G unique? If not, give a different one.

4. (Bonus question 15pts, Due 11:59pm Dec 9) Implement an algorithm to produce all connected components for a given graph G. Assume the input graph is in a given input file, in the following format. Write your output on the standard output. Make sure your program should be able to read in graphs in various sizes. Cite properly on the resources you refer to.

For the graph



Your input file graph.txt should look like:

0 1 1 1 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
0 0 0 0 0 1
0 0 0 0 1 0

Your output on screen should look like:

G has 2 connected components

0 1 1 1
1 0 0 0
1 0 0 0
1 0 0 0

0 1
1 0