CS423: Networks

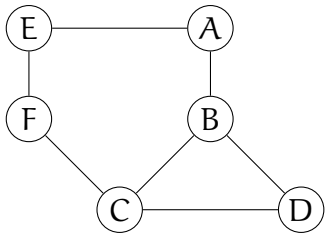
Assignment 1

Answer all 4 questions, each is worth 1 point. 1 additional point may be awarded for a particularly nice answer. A maximum of 5 points can be earned from this assignment.

Deadline for submission: Thursday, **February 11 at 5pm**.

1. Recall that a **shortest path** between two nodes is a path of minimal possible length. A node X is called **pivotal** for a pair of nodes Y and Z, if X lies **on every shortest path** between Y and Z (and X is different from Y and Z).

For example, in the graph on the right, node B is pivotal for the pair A, C and for the pair A, D. But B is not pivotal for the pair D, E, as one shortest path from D to E does not pass through B. The node D is not pivotal for any pair of nodes.



(a) Give an example of a graph in which every node is pivotal for at least one pair of nodes. Explain your answer.

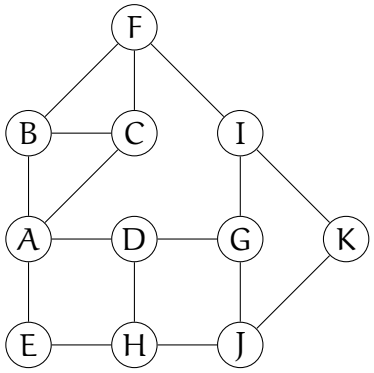
(b) Give an example of a graph in which every node is pivotal for at least two different pairs of nodes. Explain your answer.

2. The **social graph** of a node A in a (social) network is the induced subgraph on the set of friends of A. On MathSciNet (http://www.ams.org/mathscinet), pick a (local) mathematician with at least 10 friends (i.e., co-authors). Then sketch their social graph and determine the clustering coefficient.

3. Design a computer program that allows you to experimentally test the claim that, in a **random graph** on n vertices, a **giant component** becomes visible, after the graph has acquired approximately $\frac{n}{2}$ edges in random positions. (This question is asking for somewhat detailed descriptions of the individual parts of such a computer program: what data structure is used to represent vertices and edges, how can edges be chosen uniformly at random, how can a giant component be recognised, how to find the connected components of a network in the first

place, ... The question does **not** ask for an actual implementation of such a program. However, an actual implementation in your favorite programming language would be an acceptable solution.)

4. In the graph on the right, use Breadth First Search as one of the tools to determine how the flow from node B spreads out over the edges of the graph. Do the same for another twor or three nodes .... Can you compute the betweenness values of all of the edges from these data? Which edge of the graph would the Girvan-Newman method for graph partitioning remove first?



(Challenge: Write a computer program that performs the Girvan-Newman method on any given graph.)