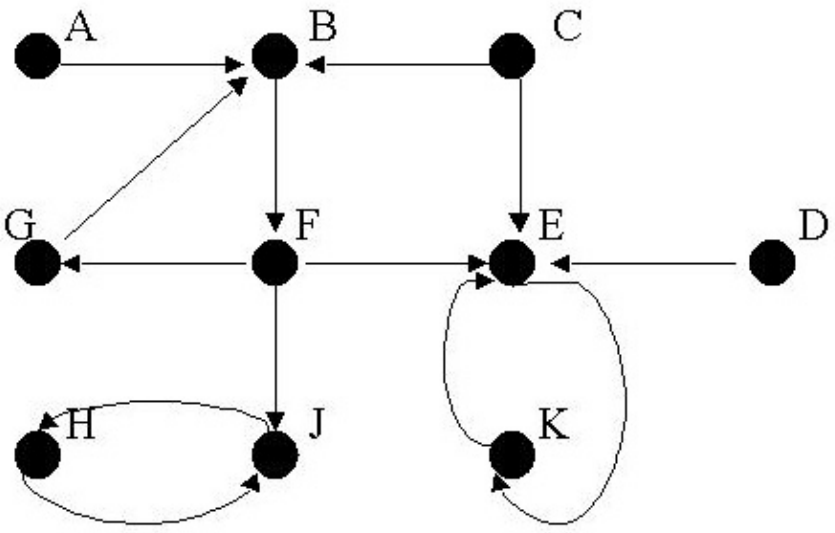
CS 170, Fall 1997
First Midterm
Professor Papadimitriou

Problem #1

(15 points) What are the strongly connected components of the directed graph shown below? (Just circle them).



- Which is the strongly connected component that will be discovered first by our algorithm (first do DFS on the reverse graph, then on the graph)? Which one will be discovered last? *As always, when DFS has a choice, it visits the lexicographically smallest node.*
- Give the DAG of the strongly connected components. Give a topological sorting of this DAG.

Problem #2

2. (15 points) Give a linear algorithm which, given a connected undirected graph, returns a node whose deletion would *not* disconnect the graph.

A. Brief Description or pseudocode

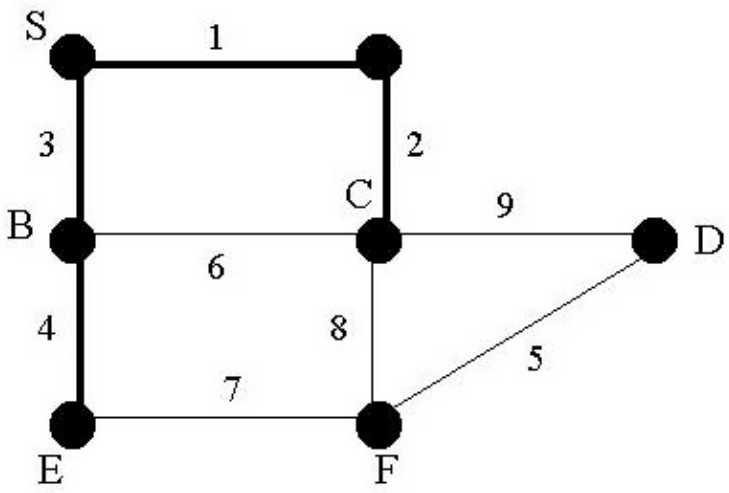
B. Justification of correctness.

C. Running time and justification.

Problem #3

(15 points) We are running one of these three algorithms in the graph below, and we have added the bold edges:

- Prim's for the minimum spanning tree, starting from *s*.
- Kruskal's (or greedy) for the minimum spanning tree.
- Dijkstra's for shortest paths from *s*.



Which edge would be added next in Prim's algorithm?

Which edge would be added next in Kruskal's algorithm?

Which node would be labeled next in Dijkstra's algorithm?

In Kruskal's algorithm with union-find, what is the maximum rank of any node before this step?

Problem #4

(10 points) We are given a directed graph with lengths on the edges. *Only one* edge hs negative length, all other edges have positive length. Give and algorithm for detecting whether this graph has a negative cycle. Your algorithm should be as fast as Dijkstra's.

A. Brief Description or pseudocode

B. Justification of correctness.

C. Running time and justification.

Problem #5

(Total of 20 points)

True or false? No explanation required, except for partial credit. Points *will* be subtracted for wrong answers, so answer only if you are reasonably certain.

- Two directed graphs G1 and G2 have the same number of nodes, G1 is a DAG, which G2 is not a DAG. Then G2 must have fewer strongly connected components than G1.
- If we add a directed edge to a directed graph, the number of strongly connected components cannot increase.

- Adding a directed edge to a directed graph may decrease the number of strongly connected components by at most one.
- If all edge weights are distinct, the minimum spanning tree is unique.
- If all edge weights are distinct, the *second best* spanning tree is unique.
- If all edge weights are distinct, the *maximum* spanning tree is unique.
- In the union-find data structure you do not need "child" pointers.
- In a *d*-heap, **deletemin** is cheaper than **insert.**
- Consider a weighted undirected graph, a cycle in it, and an edge in the cycle whose weight is larger than the weight of any other edge in this cycle. Then this edge will never be a part of the minimum spanning tree.
- Consider a weighted undirected graph, a cycle in it, and an edge in the cycle whose weight is smaller than the weight of any other edge in this cycle. Then this edge will always be a part of the minimum spanning tree.
- For data structure, O(*log n)* worst-case time *O(log n)* amortized time, but not vise-versa.
- Fibonacci and Leonardo di Pisa were the same person.
