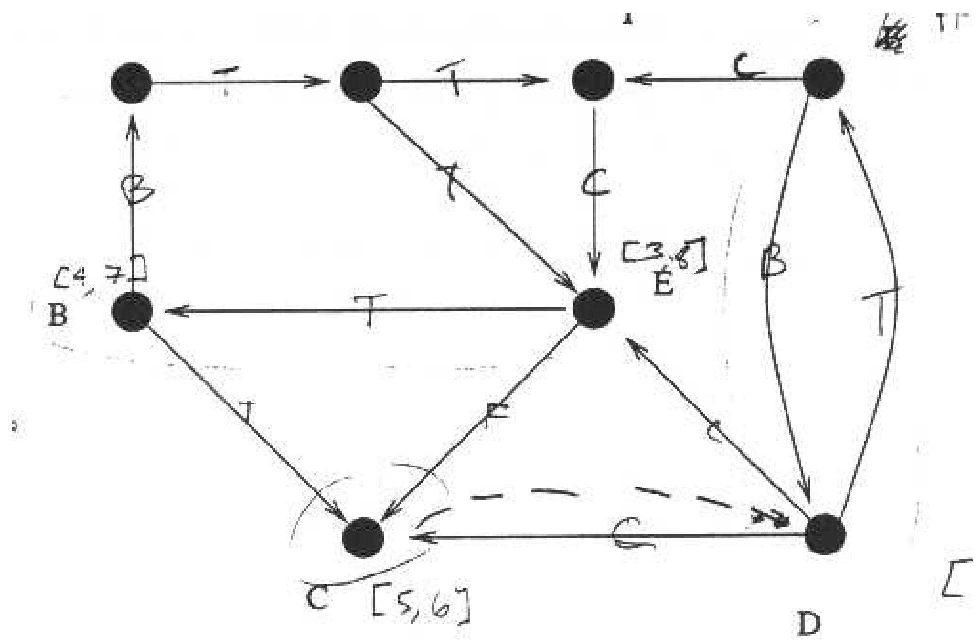
CS 170, Fall 1998
Midterm 1

Problem #1

1. (15 points) What are the strongly connected components of the directed graph shown below? (Just circle them).



Perform depth-first search on the graph shown above. For each vertex show the pre-order and post-order numbers, and label each edge with T, F, B, C depending on whether it is a tree, forward, back or cross edge. (As always, when in DFS we have a choice, we always select the alphabetically first vertex.)

Can you add an edge to this graph to make it strongly connected?

Problem #2

2. (15 points) We are given a tree T = (V,E) rooted at vertex r is an element of V. Recall that vertex u is an ancestor of vertex v in the rooted tree, if the path from r to v in the tree goes through u.
We wish to preprocess the tree by associating numbers with the vertices, so that queries of the form "is u an ancestor of v?" can be answered in constant time. The preprocessing should take linear time. (Hint: Think depth-first search.)

A. Brief description or pseudocode for the preprocessing

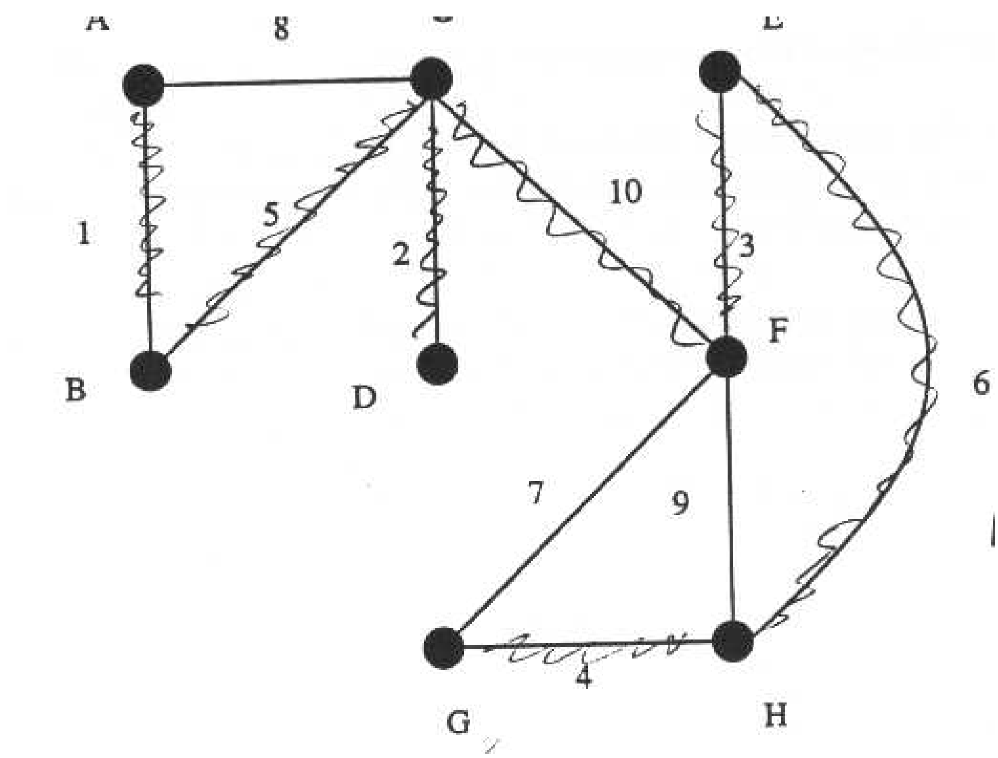
B. Brief description or pseudocode for answering a query

C. Justification of correctness (for both).

D. Running time and justification (for both).

Problem #3

3. (15 points) We wish to find the minimum spanning tree of the graph below.



Give the order in which edges will be added to the MST by Prim's algorithm. Start the algorithm from vertex A.

Repeat for Kruskal's algorithm.

Show the final configuration of the UNION-FIND tree at the end of Kruskal's algorithm. (UNION-FIND means of course "with path compression". In case of ties, always make the alphabetically larger node the root.)

Problem #4

4. (15 points) In Internet routing, the delays on lines are usually negligible, but the delays on routers may be substantial. This motivates a different kind of shortest path problem:

We are given a graph with weights on its vertices. The weight of a path is the sum of the weights of the vertices on the path --including the endpoints. Give an algorithm for finding a shortest path between two nodes s and t. Your algorithm should run as fast as Dijkstra's.

A. Brief description or pseudocode

B. Justification of correctness.

C. Running time and justification.

Problem #5

5. (Total of 30 points)

True or false? Circle the correct answer. No explanation required. Points will not be subtracted for wrong answers.

T F If a directed graph G = (V, E) has |V| strongly connected components, then it has no cycles.
T F If a directed graph G = (V, E) has |V| - 1 edges, then it is not strongly connected.
T F If a directed graph G = (V, E) has 2|V| edges, then it must have a cycle.
T F Every single operation (union or find) in UNION-FIND takes O(log n) steps.
T F Every operation in UNION-FIND takes O(log* n) steps.
T F At least half the operations in UNION-FIND take O(log* n) steps.
T F f(n) = O(s(n)) and g(n) = O(r(n)) implies f(n) - g(n) = O(s(n) - r(n)).
T F Prim's algorithm for the minimum spanning tree problem works correctly when the graph has negative edges.
T F The shortest path from s to v in a weighted directed graph G = (V, E) remains unchanged if the weight of every edge in G is increased by the same positive constant c.
T F The shortest path from s to v in a weighted directed graph G = (V, E) remains unchanged if the weight of every edge in G is multiplied by the same positive constant c.
T F There is a polynomial time time algorithm for finding the longest simple path in a tree with weights on the edges.
T F If the edge weights in a directed graph are all positive and equal, shortest paths from s to every vertex can be computed in linear time.
T F If T and T' are both minimum spanning trees in G(V, E), then the heaviest edge in T must have the same weight as the heaviest edge in T'.
T F In the DFS on a DAG, the vertex with the lowest postorder number is necessarily a sink.
T F "Algorithm" is a word in Arabic that originally meant "tent-pitching method."
