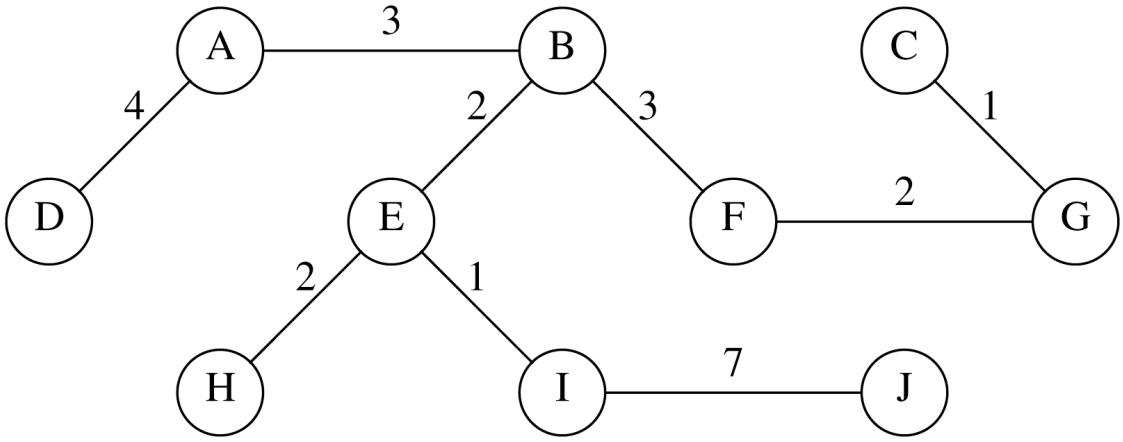
Homework 7 solution

9.15 (10 pts) The solution is not unique. One possible solution is:

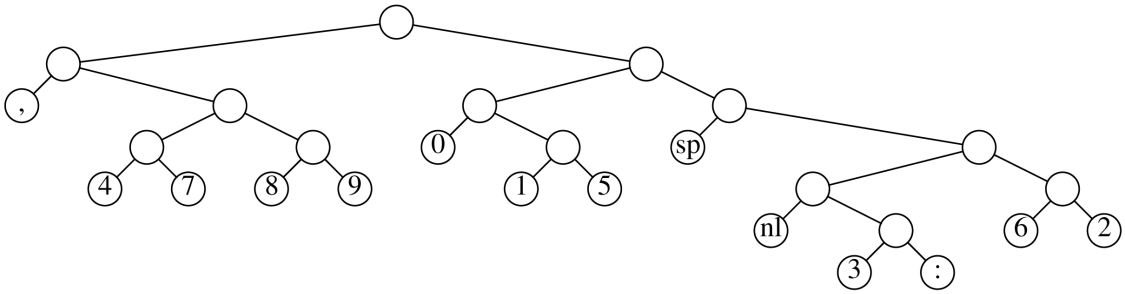


9.16 (3 pts)
Both work correctly. The proof makes no use of the fact that an edge must be nonnegative.

9.20 (7 pts)
Since the minimum spanning tree algorithm works for negative edge costs, an obvious solution is to replace all the edge costs by their negatives and use the minimum spanning tree algorithm. Alternatively, change the logic so that < is replaced by >, *min* by *max*, and vice versa.

9.39 (10 pts)
Use a depth-first search, marking colors when a new vertex is visited, starting at the root, and returning false if a color clash is detected along a back edge.

10.3 (10 pts)



10.6 (10 pts)
Maintain two queues, Q_1 and Q_2. Q_1 will store single-node trees in sorted order, and Q_2 will store multinode trees in sorted order. Place the initial single-node trees on Q_1, enqueueing the smallest weight tree first. Initially, Q_2 is empty. Examine the first two entries of each of Q_1 and Q_2, and dequeue the two smallest. (This requires an easily implemented extension to the ADT.) Merge the tree and place the result at the end of Q_2. Continue this step until Q_1 is empty and only one tree is left in Q_2.