MATH 3012B **Name:** _____
Quiz #3 11 November 2008

1. (5 points) A new local bank is being created and will establish a headquarters H, two branches B_1 and B_2, and four ATMs A_1, A_2, A_3, and A_4. They need to build a computer network such that the headquarters, branches, and ATMs can all intercommunicate. Furthermore, they will need to be networked with the Federal Reserve Bank of Atlanta, F. The costs of the feasible network connections (in units of \$10,000) are listed below:

HF	80	HB_1	10
HB_2	20	B_1B_2	8
FB_1	12	FA_1	20
B_1A_1	3	A_1A_2	13
HA_2	6	B_2A_2	9
B_2A_3	40	A_1A_4	3
A_3A_4	6		

The bank wishes to minimize the cost of building its network (which must allow for connection, possibly routed through other nodes, from each node to each other node), however due to the need for high-speed communication, they **must** pay to build the connection from H to F as well as the connection from B_2 to A_3. Give a list of the connections the bank should establish in order to minimize their total cost, subject to this constraint.

Solution: We can just run Kruskal's algorithm (avoid cycles) here, starting with the spanning forest that contains only the two edges HF and B_2A_3. This will result in a spanning tree of minimum weight among all those that contain these two edges, since Lemma 10.3 guarantees that adding an edge of minimum weight with one end in a component C of a given spanning forest and the other end not in C will allow you to find a minimum weight spanning tree containing your initial spanning forest. To do this, we sort the edges (other than HF and B_2A_3):

B_1A_1	3	HB_1	10
A_1A_4	3	FB_1	12
HA_2	6	A_1A_2	13
A_3A_4	6	HB_2	20
B_1B_2	8	FA_1	20
B_2A_2	9		

We are able to add B_1A_1 without creating a cycle. We are then able to add A_1A_4 without creating a cycle. We can then add A_3A_4 without creating a cycle. Next, we add HA_2 and do not create a cycle. We next consider B_1B_2, but we must reject it, as it would create the cycle $(B_1, A_1, A_4, A_3, B_2)$. Adding B_2A_2 is permissible, however, which means the list of edges we use is

$$FH, B_2A_3, B_1A_1, A_1A_4, A_3A_4, HA_2, B_2A_2.$$

(We know we can stop here, was we have eight vertices, so a spanning tree has seven edges.)

As an alterntive solution, we can think about running Prim's algorithm (build tree) in a modified form where we require that whenever we've added one of the four vertices incident

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community.

with the two edges we're required to have in our tree, we next must add the other end of that edge. Thus, here we would (suppose we start with H as our root) first add HF. Then the lightest edge to add is HA_2. We next add A_2B_2, as it is lightest among those from our tree to the rest of the graph. Now we don't get to choose the lightest edge, we **must** add edge B_2A_3. We can now continue, and see that we add A_3A_4, followed by A_4A_1, and finally A_1B_1.