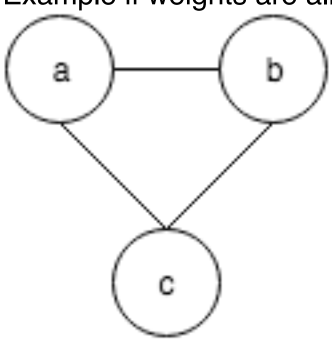
Zijian Gao
26319560
4/7/2016
CS311 HW4

23.1-7.)

Proof that if edge weights must be positive, any subset of edges that connects all vertices and has minimum total weight must be a tree:

If all edge weights are positive, there exist no negative-weight cycles. Since there are no negative-weight cycles, no edge that exists in some minimum weight subset of edges that connects any vertex may form a cycle, since the existence of a negative cycle in that set would mean that the total weight of those edges could approach negative infinity by navigating through the cycle an infinite amount of times. Positive cycles would not be contained in the arbitrary minimum subset because it is only possible for the total edge weight of the set of edges to increase by iteratively traversing a positive-weight cycle, and since it is stated to be negative, no positive cycles exist, either. It is given that the set of edges connect all of the vertices, which is some connected graph by definition. Because that graph cannot have cycles, it is a tree.

Example if weights are allowed to be nonpositive:



Let all edge weights in the above diagram be 0. In this case, the minimum subset of edges has a total weight of 0, and is (a,b),(b,c),(a,c). This set of edges does have a minimum edge weight, but does not form a tree because they form a cycle, and that violates the tree property of being acyclic. I note however that there do exist subsets that do not contain cycles and are minimum as well, such as (a,b),(b,c), but one counterexample shows that the previous conclusion does not apply if weights are allowed to be nonpositive

23.2-1.)

Given: some desired minimum spanning tree T of some graph G
Proof that for each T of G, there is a way to sort edges so that Kruskal's Algorithm returns T:

Since it is known that T is a minimum spanning tree, it is equivalent in total weight to all other minimum spanning trees of G. When sorting the edge weights in G, ensure that edge *e* precedes all edges that are not in T with edge weight *w(e)*. This would guarantee T is output, since once the first(correct) edge e is added to the spanning tree by the algorithm, all other edges with the same weight as e will be disregarded by the algorithm since those edges will no longer be disjoint from the tree containing e.