1. In an undirected weighted graph, recall that the width of a path is the minimum weight of one of its edges, and define the width of a cut to be the maximum weight of one of its edges. Prove that, for every two vertices s and t, the width of the widest path from s to t equals the width of the narrowest cut separating s from t (the cut with the smallest possible width). Hint: Use the fact that the maximum spanning tree contains the widest path to find a cut with this width. You may assume that no two edges have equal weights if it simplifies your proof.

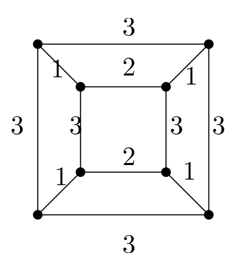
Solution: Let T be a maximum spanning tree and P be the path in T that connects s and t. Removing the minimum-weight edge e in P divides the tree T into two components, one containing s and the other containing t. Let C be the cut between these components.
Since P is a widest path between s and t and P has the same width as e, it suffices to prove the following two claims to prove the result:
(1) the width of cut C equals the width of e, and
(2) all cuts between s and t have width greater than or equal to the width of e.

Proof of (1): No edge e' in C other than e is in T. Therefore, all these edges e' must be narrower or have the same width as e. Hence e and C have the same width.

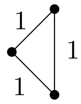
Proof of (2): consider another cut C' between s to t. This cut must have an edge e' that is on path P. By definition of cut width, width$(C') \geq$ width(e'), and by definition of path width, width$(e') \geq$ width(e). So width$(C') \geq$ width(e).

2. The graph of a cube has eight vertices and twelve edges. Find weights for these edges such that Boruvka's algorithm takes three iterations to construct the minimum spanning tree of the graph.

Solution:



3. (163 only): Find a weighted undirected graph G, and a minimum spanning tree T of G, such that at least one of the minimum weight edges in G does not belong to T. Explain why this does not violate the cut property of minimum spanning trees described in class.

Solution: 

In this graph, any two of the three edges forms a minimum spanning tree. The cut property is not violated because the omitted edge does not have width strictly less than the width of the edge in the cut.

(265 only): Let G be a graph in which all edge weights are positive. Describe a method for constructing a spanning tree that maximizes the product of the edge weights (instead of their sum). Explain why your method is correct.

Solution: We claim that a spanning tree with maximum weight by product of edge weights also has maximum weight by sum: the proof of the min cut property holds even when the sum is turned into a product. Therefore, any algorithm that computes the maximum spanning tree will work.

4. (163 only): Describe how to modify Kruskal's algorithm to compute a maximum spanning tree instead of a minimum spanning tree.

Solution: Consider the edges in decreasing order of weight rather than increasing.

(265 only): Describe how to modify the maximum-spanning-tree version of Kruskal's algorithm to compute a widest cycle in a given weighted undirected graph. (That is, among all simple cycles, we want the one whose lightest edge is as heavy as possible.)

Solution: Modify Kruskal's algorithm so that it considers edges in decreasing order. The first cycle it detects must be the widest cycle, as all other cycles will contain a lighter edge than than the first cycle.