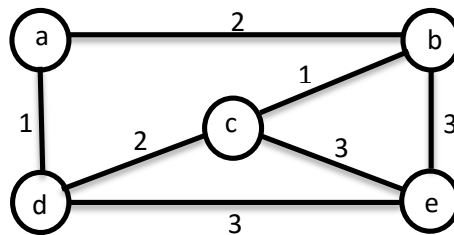


CSC 226 FALL 2016
ALGORITHMS AND DATA STRUCTURES II
ASSIGNMENT 2 - WRITTEN
UNIVERSITY OF VICTORIA

1. Suppose G is a connected graph with distinct positive integer edge weights. Prove that G has exactly one minimum spanning tree T . You may invoke the cut property and/or the cycle property if needed.
2. Modify the statement of the cut property so that it is true, even for the case of edge weights that are not necessarily distinct. Prove that your modified statement is true when edge weights may be repeated.
3. Consider the following graph $G = (V, E)$, where each edge $e = (u, v)$ has a weight $w(u, v)$, $V = \{a, b, c, d, e\}$ and $E = \{(a, b), (b, c), (c, d), (d, a), (b, e), (c, e), (d, e)\}$. The weights on the edges are:



We define the *profile* of a graph G to be the list of its edge weights in non-decreasing order. For example, in the graph G above, the profile is 1,1,2,2,3,3,3,3. The profile of an MST T is the list of its edge weights in non-decreasing order.

Find every MST T for the graph G above: for each T , list the edges in alphabetical order, give the weight of T and its profile p . We know that the weights of all the MSTs, by definition, must be the same. Based on the results can you make a conjecture about profiles of MST's?

4. In the *Algorithms, 4th Edition* textbook by Sedgwick, in Chapter 1.5 he develops an implementation of the union-find data structure using an integer array of size N for N vertices called the site-indexed $\text{id}[]$ array. Show the contents of the $\text{id}[]$ array along with the corresponding forest of trees representation (see example on page 225) for each input pair 9-0 3-4 5-8 7-2 2-1 5-7 0-3 4-2 when you use the weighted quick-union algorithm (page 228). This algorithm corresponds to the union-by-size algorithm discussed in lecture. (This is actually exercise 1.5.3 in the text.)
5. For that matter, why don't you do exercise 1.5.9 from the textbook while you're at it. Draw the tree corresponding to the $\text{id}[]$ array given below:

i	0	1	2	3	4	5	6	7	8	9
id[i]	1	1	3	1	5	6	1	3	4	5

Can this be the result of running weighted quick-union? Explain why this is impossible or give a sequence of operations that results in this array.