

CSE 326
Autumn 2005
Assignment 7
Due 11/23/05

For all algorithm and data structure design problems please provide elegant pseudocode and an adequate explanation of your methods. It is often helpful to include small examples demonstrating the method. Put your name at the top of each sheet of paper that you turn in.

1. Modify Dijkstra's algorithm so that if there is more than one minimum length path from the source to a vertex, one with the fewest number of edges is chosen for the previous pointers. Hint: for each vertex more than the minimum length path known so far must be maintained during the algorithm.
2. Consider the following sequence of disjoint union / find operations: union(1,2), union(2,3), union(3,4), union(4,5), union(5,6), union(6,7), union(7,8), union(9,10), union(11,12), union(13,14), union(15,16), union(1,10), union(1,12), union(14,15), union(1,16). In this problem we don't assume that the inputs to union are roots, so that two find operations are performed during the union to find the roots before pointing one root to another. Show the resulting up tree after these operations for each case below. In each case count the number of nodes visited in all the find operations. In the case of path compression some nodes are visited twice.
 - (a) There is no path compression on the finds and the root of the first argument points to the root of the second argument.
 - (b) There is path compression on the finds and the root of the first argument points to the root of the second argument.
 - (c) There is no path compression on the finds and weighted union is used.
 - (d) There is path compression on the finds and weighted union is used.