CST 428/528

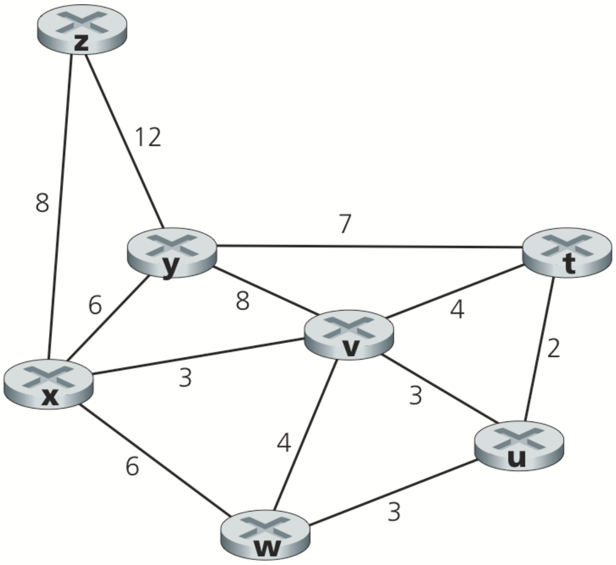
Instructor: Anand Seetharam

Programming Assignment 4: Dijkstra's Algorithm and Traceroute

In this programming assignment, you will implement Dijkstra's algorithm in C++ to compute the least-cost path from one source to all destinations. For this assignment, you can refer to the pseudo code give in Page 367 of Kurose- Ross textbook.

Consider the abstract graph model of a computer network (e.g., Figure 4.27 of the textbook). One standard way to input a graph is to use an adjacency matrix. Assume that there are N nodes in your network/graph. The matrix is of size N*N. The value of the (i,j)th entry (i.e., ith row and jth column) denotes the edge weight between nodes i and j. If two nodes have no edges between them, then assign the edge weight to a large number. For this topology, your program should be able to print out a table (e.g., Table 4.3) demonstrating the running of Dijkstra's algorithm. The program should be able to take as input (i.e., source) any node in the network and then output the above table.

a) Test that you code works correctly by testing it against Figure 4.27.
b) Test your program for the topology given below.



c) Test your program for other edge weights for the same topology (use the interactive version of the book) http://gaia.cs.umass.edu/kurose_ross/interactive/dij.php

What to hand in?

1. Submit code in C++ with inline documentation

2. A design document describing how to compile your code, how to execute it and what design decisions you made.
3. Tables for parts a), b) and c) printed out from your program which demonstrates that your program runs correctly.
4. Execute the code to me/TA to demonstrate that it works correctly

Grading Criteria (80 points)

In-line documentaton – 10 points
Code compiles and executes correctly – 60 points
Design document – 10 points

Traceroute (20 points)

Perform a Traceroute between source and destination on the same continent at three different hours of different days. You need to provide screenshots to show that you have done all these experiments. Please do not try websites such as google.com, amazon.com etc as these websites make extensive use of CDNs and you will not get useful results.

a. Find the average and standard deviation of the round-trip delays at each of the three hours.
b. Find the number of routers in the path at each of the three hours. Did the paths change during any of the hours?
c. Try to identify the number of ISP networks that the Traceroute packets pass through from source to destination. Routers with similar names and/or similar IP addresses should be considered as part of the same ISP. In your experiments, do the largest delays occur at the peering interfaces between adjacent ISPs?
d. Repeat the above for a source and destination on different continents. Compare the intra-continent and inter-continent results.