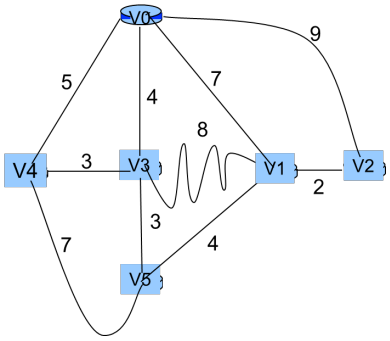
Homework 10, Due Date*: 11:59pm 5/1/2015, Cutoff Date: 11:59pm 05/03/2015**
Submission: Upload all source code (.java) files and all .txt files to Blackboard,
and java programs to EC2 instances (optional if you are using standard jdk; required if you are using anything special)
***Late penalty will apply for past-due late submission; ****Submission will NOT be accepted after the cutoff deadline**

Write a program to simulate the reaction at **router V_0** to the change in local link cost or receiving a message from its neighbor. (**Hint**: for testing, you may want to use the network given below or anyone with a similar complexity instead of the simple network with only three routers x, y, and z. You may want to use the given topology to calculate the entries in Distance Vector $\mathbf{D_0}$, Link Vector $\mathbf{L_0}$, and Distance Vectors of EACH neighbor of **router V_0**, which are the user inputs needed in Step 2)



- The routers in the network are labeled as V0, V1, V2, …, etc
- Your routing program needs to
  1. Display a prompt message to ask the user to input the total number of routers, ***n***, in the network. Validate ***n*** to make sure that it is greater than or equal to 2.
  2. Use ***.txt files*** to get the stabilized information maintained at router V0 before it deals with any change (event).
     a. Read from the ***cost.txt*** file to get the link costs to all neighbors of node V0. Each line is formatted as
        <neighboring node id> <link cost>
     where the first input needs to be validated to be between 1 and ***n – 1*** and the second input needs to be validated to be positive. For example, for the topology above, the lines are
        4 5
        3 4
        1 7
        2 9
     b. Read from the ***source_vectors.txt*** file to get the distance vector ***D0*** and link vector ***L0***. The format is
        <D0[0]> <D0[1]> <D0[2]> …
        <L0[0]> <L0[1]> <L0[2]> …
     For example, for the topology above, the two lines are
        0 7 9 4 5 7
        0 1 2 3 4 3
     c. Read from the ***neighbor_vectors.txt*** file to get the distance vectors of its neighbors. Each line is formatted as
        <x> <Dx[0]> <Dx[1]> <Dx[2]> …
     For example, for the topology above, the lines are
        1 7 0 2 7 10 4
        2 9 2 0 9 12 4
        3 4 7 9 0 3 3
        4 5 10 12 3 0 6
  3. Ask the user to select one of the following two events to continue:
        Event 1: a change in local link cost to a neighbor of router V0
        Event 2: receiving a distance vector message from a neighbor of router V0
     a. If the user select Event 1, display the following two prompt messages to ask for user inputs,
        The index of this neighboring router:
        The new link cost to this neighboring router:
     b. If the user select Event 2, display the following first prompt message once and second prompt message ***n*** times to ask for user inputs,
        The index of the neighbor from which the distance vector message is received:
        The new least cost from this neighbor to router Vj, Dx(j):
     where j = 0, 1, 2, …., n – 1 in sequence and x is the index of this neighbor.
  4. Implement the **Distance Vector algorithm** to (1) recomputed distance vector $\mathbf{D_0}$ and link vector $\mathbf{L_0}$ at **router V0** according to the user inputs in Step 6 and (2) determine whether to notify neighbors. **(more on next page!)**

a. If there is no need to notify neighbors, display
   There is no need to notify any neighbor!
b. If there is a need to notify neighbors, display
   The list of neighbors to be notified
   The list of the entries in the distance vector D0 to be sent to all the above neighbors
   The list of the entries in the link vector L0 although L0 is not going to be sent out

5. Display a prompt message to ask the User whether to input a new Event of change or receiving. If yes, repeat step 6 and step
6. Otherwise, terminate this program.