Bookmark

LAB 2 - TASK 1 (EXTERNAL RESOURCE) (1.0 points possible)

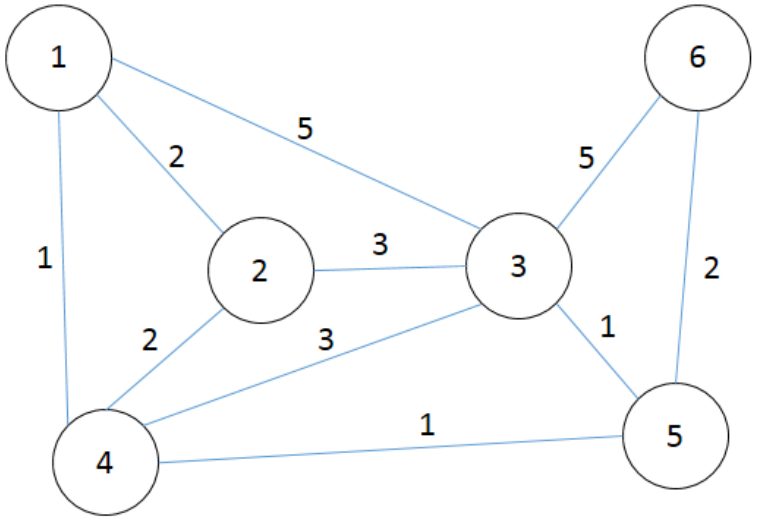
Lab 2 - Task 1

Consider the situation where you have completed your lab task for ELEC1200.3x and have hit the ' button. In order for you to receive credit for your work, the packets containing your solution must b edX server for grading. However, your computer is not directly connected with the edX server. As a packets must pass through a sequence of nodes before it arrives at the edX server. The process b node in the network takes incoming an packet and passes it to the next node along the path toward destination is known as forwarding.

In this task, we will learn how nodes use the information stored in their routing tables to perform for how packets can be routed globally through the network (from source to destination) using only loc (forwarding).

INSTRUCTIONS

Consider the simple network shown below consisting of N = 6 nodes, indexed from 1 to 6. Each lin nodes is labelled by the cost (usually interpreted as delay) associated with taking that link. Nodes t connected by links are not directly connected to each other.



The MATLAB code below simulates the process by which packets move from source to destination network. In order to perform forwarding, each node in the network maintains a routing table. In our represent each routing table as a matrix with N rows and two colums, where N is the number of nod network (6 in this example). The entire set of routing tables for all nodes in the network is represen array **RT**, where the cell **RT{i}**, contains the routing table for node **i**.

For example, the routing table of the first node is given by:

```
RT{1} =
     0     0
     2     2
     4     4
     4     1
     4     2
     4     4
```

The **m**-th row of the routing table matrix contains the information required to forward the packet to first element **RT{1}(m,1)** indicates the node to which the packet needs to be forwarded to. If the pa already at its destination, then this entry is 0. The second element **RT{1}(m,2)** shows the total cost node **m**.

Suppose node 1 receives a packet whose destination is node "**5**". Node looks at the **5**th row of the **RT{1}(5,1)=4** , node 1 passes this packet out along the link leading to node 4. The cost to reach n **RT{1}(5,2)=2**. The entry **RT{1}(1,1)=0** indicates that the packet has arrived at its destination (node forwarding is needed. The entry **RT{1}(2,1)=2** indicates that if the packet is destined for node 2, it on the link connecting node 1 to node 2. Thus, nodes 1 and 2 are directly connected (neighbors), of the link between them is **RT{1}(2,2)=2**. Similarly, **RT{1}(4,:) = [4 1]** indicates that nodes 1 and 4 a connected with cost 1. On the other hand, even though nodes 1 and 3 are directly connected, the **(3,1)=4** tells node 1 that packets destined for node 3 should first goto node 4. This is because the between nodes 1 and 3 is relatively slow (cost = 5), whereas the cost to get to node 3 through nod **RT{1}(3,1)=4**.

The MATLAB code below simulates the iterative process by which a packet gets from the node **sta** the node **dest_node** by forwarding at each node. We assume that the routing tables have alread computed and are stored in the cell array **RT**. We start the process by setting the current node of t

computed and are stored in the cell array **RT**. We start the process by setting the current node of t
the starting node with the line of code

curr_node = start_node

Each iteration proceeds as follows, the current node looks at its routing table, **RT{curr_node}**, to c
node to pass the packet to. The code then passes the packet to that node by updating the value c
curr_node to that node for next iteration. The iteration terminates when the **curr_node** is equal t
when the number of loops exceeds the **timeout**. Note that this is because we set **RT{i}(i,1)=0**, so th
curr_node=0 indicates that the packet has arrived at its destination.

The code simulates multiple packets going from a variety of start nodes to a variety of destination r
paths taken by these packets are stored in the matrix **path**.

Your task is to revise the code between the lines

```
    % % % % Revise the following code   % % % %
```

and

```
    % % % % Do not change the code below % % % %
```

so that the variable **curr_node** at the end of each iteration is updated so that the packet has beer
to the next node. Do not change other parts of the code.

Your Solution

Reset MATLAB Documentation (https://www.mathworks.

```
 8 load RT.mat
 9 disp(RT{1});
10 disp(RT{2});
11 disp(RT{3});
12 disp(RT{4});
13 disp(RT{5});
14 disp(RT{6});
15 % init the current node, the time and the path
```

Output

```
0     0
      2     2
      4     3
      4     1
      4     2
      4     4

      1     2
      0     0
      3     3
      4     2
      4     3
      4     5

      5     3
      2     3
      0     0
      5     2
      5     1
      5     3

      1     1
      2     2
      5     2
      0     0
      5     1
      5     3

      4     2
      4     3
```

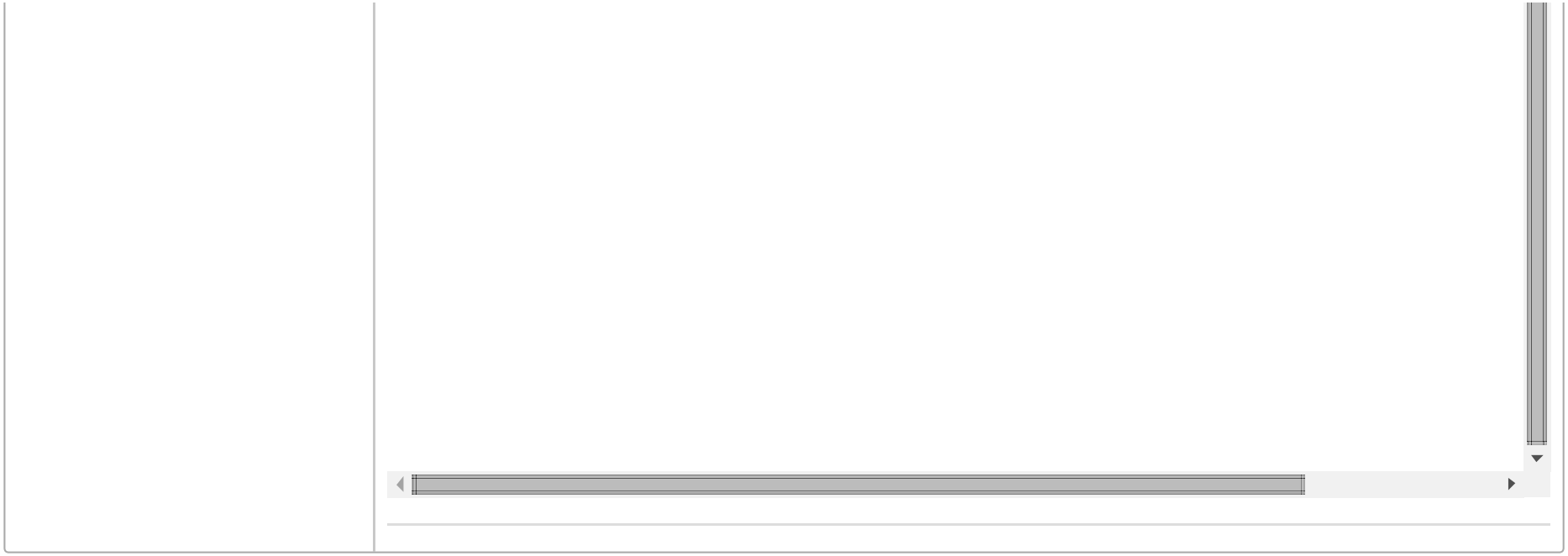
```
     4     3
     3     1
     4     1
     0     0
     6     2

     5     4
     5     5
     5     3
     5     3
     5     2
     0     0


ans =

     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     2     2     2     2     2
     2     2     2     2     2
     2     2     2     2     2
     2     2     2     2     2
     3     3     3     3     3
     3     3     3     3     3
     3     3     3     3     3
     4     4     4     4     4
     4     4     4     4     4
     5     5     5     5     5
```

Assessment Tests




Download on the
App Store


ANDROID APP ON
Google play