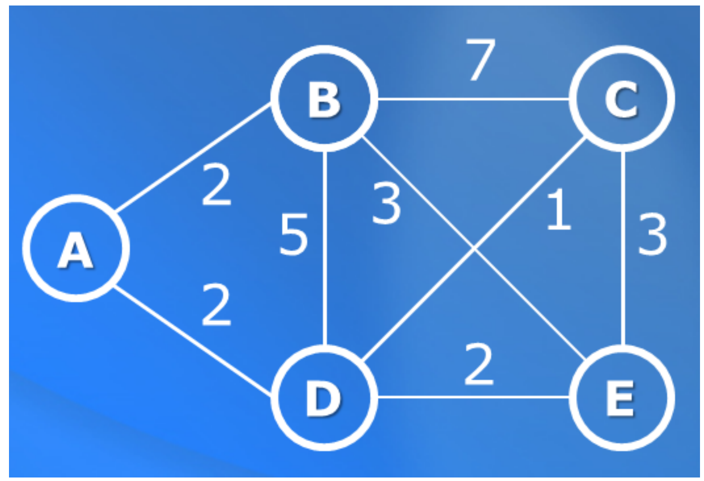
The figure below shows a network consisting of five nodes. Links between nodes are labelled by costs. Use this network to answer Questions 1-3.



Section 2 Question 1

(2/2 points)

Suppose that Node B in a network has an initial routing table given by

(A,A,2), (B,-,0), (C,C,7), (D,D,5), (E,E,3)

The routing table is given by triples with the form (dest, link, cost), where

- Topic 7: The Application Layer
- Topic 8: Course Review
- Final Exam

Final Exam
Final Exam due Feb 22, 2016 at 15:30 UTC

- MATLAB download and tutorials

- "link" indicates the next hop (the neighbor at the end of the link that should be taken by the packet) on the way from A to node "dest". An unknown or unnecessary link is indicated by "-".
- "cost" is the total cost to get to the destination node. A cost of "inf" indicates that the node does not know how to reach the corresponding destination.

Suppose that Node B knows that it can reach all nodes (A,C,D,E) directly with costs (2,7,5,3) respectively.

Advertisements from node X have the form of pairs (dest, cost), where "cost" is the cost to reach node "dest" from node X.

Suppose Node B receives the following advertisement from Node D: (A,2), (B,5), (C,1), (D,0), (E,2)

What is the routing table after the distance vector algorithm integrates this advertisement?

- ( ) (A,A,2), (B,-,0), (C,D,5), (D,A,4), (E,E,3)
- ( ) (A,D,7), (B,-,0), (C,D,6), (D,D,5), (E,D,7)
- (•) (A,A,2), (B,-,0), (C,D,6), (D,D,5), (E,E,3) ✔
- ( ) (A,A,2), (B,-,0), (C,C,7), (D,D,5), (E,E,3)

Suppose that after Node B has integrated the advertisement from D, it receives the following advertisement from Node A: (A,0), (B,2), (C,inf), (D,2), (E,inf)

What is the routing table after the distance vector algorithm integrates this advertisement?

- (x) (A,A,2), (B,-,0), (C,D,6), (D,A,4), (E,E,3) ✔
- ( ) (A,A,2), (B,-,0), (C,C,7), (D,D,5), (E,E,3)
- ( ) (A,A,2), (B,-,0), (C,A,5), (D,A,4), (E,E,3)
- ( ) (A,A,2), (B,-,0), (C,A,inf), (D,A,4), (E,A,inf)

You have used 1 of 1 submissions

Section 2 Question 2

(3/3 points)

The table below shows the partial computation of Dijkstra's algorithm to find the best path to each node from node B for the network above.

Step	N'	A		C		D		E	
		D	p	D	p	D	p	D	p
0	B	B	2	B	7	B	5	B	3
1	BA			x_1	y_1	A	4	B	3
2	BAE			x_2	y_2	A	4		
3	BAED			x_3	y_3				
4	BAEDC								

Give the value for each of the unknown quantities, marked in yellow.

x_1?

B ▼

y_1?

7 ▼

x_2?

E ▼

y_2?

6

x_3 ?

D

y_3 ?

5

You have used 1 of 1 submissions

Section 2 Question 3

(1/1 point)

After the computation by Dijkstra's algorithm above, Node B computes a forwarding table, which indicates how incoming packets should be directed.

Assume that the forwarding tables below are given in the form of pairs (destination node, next hop), where “next hop” is node at the end of the link leaving node B that should be taken to reach the “destination node” with least cost. If the packet is destined for B or node B does not know how to reach the destination node, then "next hop" is set to "-". For example, if all packets except those destined for node B should be directed out the link towards node C, then the fowarding table should be (A,C), (B,-), (C,C), (D,C), (E,C).

Which of the following is the forwarding table for node B?

- (A,A), (B,-), (C,A), (D,A), (E,A)
- (A,A), (B,-), (C,A), (D,A), (E,E) ✔
- (A,A), (B,-), (C,D), (D,A), (E,E)
- (A,A), (B,-), (C,C), (D,D), (E,E)

You have used 1 of 5 submissions

MATLAB INSTRUCTIONS

In this task, you will use the Dijkstra's algorithm to compute the shortest path from a given source node to all other nodes of the network. Dijkstra's algorithm consists of two steps. In the first step, for each destination node it computes the cost of the shortest path from the source to that node and its predecessor on the shortest path (i.e. the node visited right before arriving at the destination if the shortest path is followed). In the second step, it uses the result of the first step to compute the actual path by a process called backtracking. Your task is to implement the first step, i.e. the computation of the cost of the shortest path and finding the predecessor node for each destination node in the network as described in Lecture 4.3.

In the MATLAB code below, the network contains **n_nodes** nodes, and is represented by the matrix **graph**, as described in task 4 of Lab 2. The code first initializes two variables, **est_cost**, and **predecessor**.

The **n_nodes** by 1 matrix, **est_cost**, contains the estimated cost of the best path from the source node (**source = 8**) to each of the nodes in the network. For the neighbors of **source**, this is initialized to the cost of the direct link. For the other nodes it is initialized to infinity.

The **n_nodes** by 1 matrix, **predecessor**, contains the predecessor node for each destination node. It is initialized to **source** for the neighbors of source, and zero otherwise, indicating that the predecessor is unknown. Recall that nodes are indexed from 1 to **n_nodes**.

Your task is to implement the code that iteratively updates **est_cost** and **predecessor** using Dijkstra's algorithm, so that they eventually contain the information required to find the best paths by backtracking. Recall from Lecture 4.3, that Dijkstra's algorithm does this by iteratively adding nodes to a set N' containing the nodes for which the best path is known. In the MATLAB code, we define instead the vector **queue**, which is the complement of N', i.e., the set of nodes for which the best path is not yet finalized.

In the implementation, the main iteration proceeds by first finding **w**, the node in **queue** that has the lowest path cost and deleting it from **queue**.

Your task is to implement the next step. In particular, for each of the nodes **n** remaining in **queue**, you should compute the cost of the path that first follows the best path to **w** (with cost **est_cost(w)**) and then goes directly from **w** to **n** (with cost **graph(w,n)**) and compare it with the current estimated cost (**est_cost(n)**). Note that the cost from **w** directly to **n** may be infinity if they are not neighbors. If the cost through **w** is smaller than **est_cost(n)**, then the code should update **est_cost(n)** and set the **predecessor(n)** to **w**.

The loop then repeats, and will eventually terminate when **queue** is empty.

Please, revise the code between the lines:

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

and

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

Do not change other parts of the code.

The remaining parts of the code implement the back tracking algorithm to find the best paths, and displays the results as a **n_nodes** by **n-nodes** matrix **path** where each column shows the sequence of nodes followed from the source to each destination. The zeros at the end of each column of path are fillers, since the path is usually shorter than **n_nodes** long. It also displays the estimated cost of each path.

Section 2 MATLAB Question

(2 points possible)

```
% the variable graph encodes the network
graph = ...
    [ 0 2 4 inf*ones(1,7);
    2 0 1 inf inf 2 inf inf 4 4;
    4 1 0 4 2 2 inf inf inf inf;
    inf inf 4 0 inf*ones(1,6);
    inf inf 2 inf 0 inf 1 inf inf inf;
    inf 2 2 inf inf 0 4 4 3 inf;
    inf inf inf inf 1 4 0 1 inf inf;
    inf*ones(1,5) 4 1 0 inf inf;
    inf 4 inf inf inf 3 inf inf 0 inf;
    inf 4 inf*ones(1,7) 0]

% number of nodes
n_nodes = size(graph,1);
```

Unanswered

graph =

0	2	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2	0	1	Inf	Inf	2	Inf	Inf	4	4
4	1	0	4	2	2	Inf	Inf	Inf	Inf
Inf	Inf	4	0	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	2	Inf	0	Inf	1	Inf	Inf	Inf
Inf	2	2	Inf	Inf	0	4	4	3	Inf
Inf	Inf	Inf	Inf	1	4	0	1	Inf	Inf
Inf	Inf	Inf	Inf	Inf	4	1	0	Inf	Inf
Inf	4	Inf	Inf	Inf	3	Inf	Inf	0	Inf
Inf	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

Destinations:

1	2	3	4	5	6	7	8	9	10

Path from 8 to any destination:

0	0	0	0	0	8	8	8	0	0
0	0	0	0	0	6	7	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

and the relative cost

Inf	Inf	Inf	Inf	Inf	4	1	0	Inf	Inf

Run Code

You have used 0 of 5 submissions