Assignment 11

Name: Rajat Rajesh Shetty
CWID: 10477484

1.

The following paths from 'y' to 'u' that do not contain any loops:

1. y-x-u
2. y-w-u
3. y-x-v-u
4. y-w-v-u
5. y-w-x-u
6. y-x-w-u
7. y-z-w-u
8. y-x-w-v-u
9. y-z-w-v-u
10. y-w-v-x-u
11. y-x-v-w-u
12. y-z-w-x-u
13. y-w-x-v-u
14. y-z-w-v-x-u
15. y-z-w-x-v-u

So, there are total 15 paths that do not form cycles from path "y" to path "u".

3.

Step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	3,x	6,x	6,x	8,x
1	xv	7,v	6,v	3,x	6,x	6,x	8,x
2	xvu	7,v	6,v	3,x	6,x	6,x	8,x
3	xvuw	7,v	6,v	3,x	6,x	6,x	8,x
4	xvuwy	7,v	6,v	3,x	6,x	6,x	8,x
5	xvuwyt	7,v	6,v	3,x	6,x	6,x	8,x
6	xvuwytz	7,v	6,v	3,x	6,x	6,x	8,x

5. Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node *z*.

Distance vector routing algorithm exchanges the information with the neighbors and works asynchronously.

According to the distance vector algorithm, any node *m* computes the distance vector using the following formulas:

$$D_m(m) = 0$$
$$D_m(n) = \min\{c(m,n) + D_n(n), c(m,n) + D_o(n)\}$$
$$D_m(o) = \min\{c(m,n) + D_n(o), c(m,o) + D_o(o)\}$$

Note: NA is used when there is no distance value.

Construct the distance vector table for node *z* from the network diagram:

	u	*v*	*x*	*y*	*z*
v	*NA*	*NA*	*NA*	*NA*	*NA*
x	*NA*	*NA*	*NA*	*NA*	*NA*
z	*NA*	6	2	*NA*	0

Now update the table with costs of all the neighboring nodes.

	u	*v*	*x*	*y*	*z*
v	1	0	3	*NA*	6
x	*NA*	3	0	3	2
z	*NA*	6	2	*NA*	0

Update the table with minimum costs using the distance vector routing algorithm:

Example: *v* to *y*, two paths are available. *v-u-y* and *v-x-y* with costs 3 and 6 respectively. So, *v-u-y* is the path with minimum cost. Hence update the table with this value.

	u	v	x	y	z
v	1	0	3	3	5
x	4	3	0	3	2
z	6	5	2	5	0

Therefore, at node *z*, the above table will be computed by the distance vector routing algorithm.

8.

Node x table

Cost to

From		x	y	z
	x	0	3	4
From	y	∞	∞	∞
	z	∞	∞	∞

Cost to

From		x	y	z
	x	0	3	4
From	y	3	0	6
	z	4	6	0

Node y table

Cost to

From		x	y	z
	x	∞	∞	∞
From	y	3	0	6

z	∞	∞	∞

		Cost to		
		x	y	z
	x	0	3	4
From	y	3	0	6
	z	4	6	0

Node z table

		Cost to		
		x	y	z
	x	∞	∞	∞
From	y	∞	∞	∞
	z	4	6	0

		Cost to		
		x	y	z
	x	0	3	4
From	y	3	0	6
	z	4	6	0

10.

A teach step, each updating of anode’s distance vectors is based on the Bellman Ford equation, i.e., only decreasing those values in its distance vector. There is no increasing in values. If no updating, then no message will be sent out. Thus,D(x) is non-increasing. Since those costs are finite, then eventually distance vectors will be stabilized in finite steps.