CS 372 **Introduction to Computer Networks**
Self-Check Exercises: Lecture 29 **Solutions**

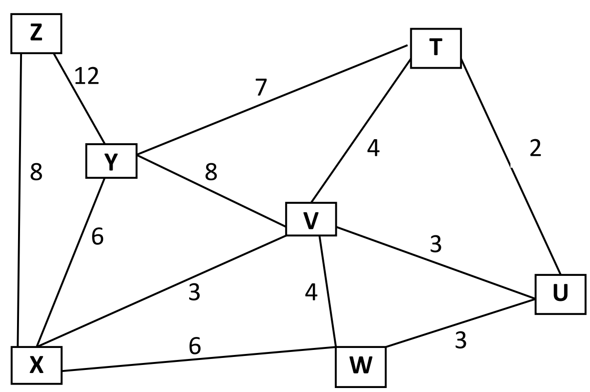
1) A ___routing algorithm___ is used to find a datagram's path through a network.

2) In a network graph…
   - "Nodes" represent ____routers____
   - Edges represent ____direct connections between routers____
   - Weights represent ____costs (speed, traffic, $$$, distance, etc…)____
   - A "shortest path" from node A to node G is ___the set of edges to traverse from A to G with the smallest sum of edge weights________________________________.

3) Once the routing algorithm is complete, what is stored in the routing table?
   A series of IP prefixes, each matched to a *single* next-hop router.

4) Given the following network represented as a graph, trace Djikstra's algorithm to determine the shortest path from *X* to all nodes. Start with destination *T*, and complete the trace for any "left-over" nodes. Show the routing table for node *X*. You may use the algorithm shown in the lecture slides (and below), or use one of the textbook algorithms. Show the contents of all arrays and variables after each iteration of the outer loop. Initialization is done for you.

```
initialize S, D, R, P;
while (!empty(S)) {
   a = node in S with D[a] a "smallest element"
         ... if tied, take smallest a;
   if(D[a] == ∞) {
      error: "no path"; exit;}
   S = S – {a};
   for (each b such that edge (a,b) exists) {
      if(b in S) {
         c = D[a] + weight (a,b);
         if(c < D[b]) {
            R[b] = R[a];
            P[b] = a;
            D[b] = c;
         }
      }
   }
}
```



Initialize **S** = {T, U, V, W, Y, Z}
Initialize tables

Dx

T	∞
U	∞
V	3
W	6
X	-
Y	6
Z	8

Rx

T	0
U	0
V	V
W	W
X	-
Y	Y
Z	Z

Px

T	0
U	0
V	X
W	X
X	-
Y	X
Z	X

First iteration: a = v   **S** = {T, U, W, Y, Z}

b = T ⇒ c = 3+4 = 7, so D[T] = 7, R[T] = v, P[T] = v
b = U ⇒ c = 3+3 = 6, so D[U] = 6, R[U] = v, P[U] = v
b = W ⇒ c = 3+4 = 7, but D[W] < 7, so no change
b = Y ⇒ c = 3+8 = 11, but D[Y] < 11, so no change

Dx

T	7
U	6
V	3
W	6
X	-
Y	6
Z	8

Rx

T	V
U	V
V	V
W	W
X	-
Y	Y
Z	Z

Px

T	V
U	V
V	X
W	X
X	-
Y	X
Z	X

Second iteration: a = U   **S** = {T, W, Y, Z} a = U because “lower” letter breaks tie

b = W ⇒ c = 6+3 = 9, but D[W] < 9, so no change
b = T ⇒ c = 6+2 = 8, but D[T] < 8, so no change

Dx

T	7
U	6
V	3
W	6
X	-
Y	6
Z	8

Rx

T	V
U	V
V	V
W	W
X	-
Y	Y
Z	Z

Px

T	V
U	V
V	X
W	X
X	-
Y	X
Z	X

Third iteration: a = W (since W and Y are tied) **S** = {T, Y, Z}

b = V $\Rightarrow$ c = 6+4 = 10, but D[W] < 10, so no change

b = U $\Rightarrow$ c = 6+3 = 9, but D[W] < 9, so no change

Dx

T	7
U	6
V	3
W	6
X	-
Y	6
Z	8

Rx

T	V
U	V
V	V
W	W
X	-
Y	Y
Z	Z

Px

T	V
U	V
V	X
W	X
X	-
Y	X
Z	X

Fourth iteration: a = Y **S** = {T, Z}

b = T $\Rightarrow$ c = 6+7 = 13, but D[Y] < 13, so no change

b = Z $\Rightarrow$ c = 6+12 = 19, but D[Y] < 19, so no change

Dx

T	7
U	6
V	3
W	6
X	-
Y	6
Z	8

Rx

T	V
U	V
V	V
W	W
X	-
Y	Y
Z	Z

Px

T	V
U	V
V	X
W	X
X	-
Y	X
Z	X

Fifth iteration: a = T **S** = {Z}

there are no nodes in **S** with an direct path to t, so no change

Dx

T	7
U	6
V	3
W	6
X	-
Y	6
Z	8

Rx

T	V
U	V
V	V
W	W
X	-
Y	Y
Z	Z

Px

T	V
U	V
V	X
W	X
X	-
Y	X
Z	X

Sixth iteration: a = Z **S** = { }

there are no nodes in **S** with an direct path to z, so no change

Dx

T	7
U	6
V	3
W	6
X	-
Y	6
Z	8

Rx

T	V
U	V
V	V
W	W
X	-
Y	Y
Z	Z

Px

T	V
U	V
V	X
W	X
X	-
Y	X
Z	X

The computed path from X to each of the other nodes is found by following the links from P[dest] to P[X] and then reversing it to get the path from node X to node dest :

T:	X → V → T	cost = 7
U:	X → V → U	cost = 6
V:	X → V	cost = 3
W:	X → W	cost = 6
Y:	X → Y	cost = 6
Z:	X → Z	cost = 8

The routing table is array Rx, above (destination node with first-hop on path).

If you prefer the textbook version:

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