HW7

1.

1st	2nd	3rd	4th	5th	6th	cost
A	-> B	-> D	-> C	-> E	-> F	10
				-> F		12
			-> E	-> C	-> F	11
				-> F		7
		-> C	-> D	-> E	-> F	11
			-> E	-> F		8
			-> F			10
	-> D	-> B	-> C	-> E	-> F	9
				-> F		11
		-> C	-> E	-> F		7
			-> F			9
		-> E	-> C	-> F		8
			-> F			4
	-> C	-> B	-> D	-> E	-> F	13
		-> D	-> E	-> F		11
		-> E	-> F			8
		-> F				10

2.

a.

D^X	W	Y
W	1	$4 + min(D^Y(W,n))$
Y	$1 + min(D^W(Y,n))$	4
A	6	10

b. $c(X,W) = 10$

$=> \ D^X(A,W) = 10 + 5 = 15 > 10 = D^X(A,Y)$

$=> \ min(D^X(A,n)) = min(10+5, 10) = 10 = D^X(A,Y)$

$=> \ new\ min\_cost\ from\ X\ to\ A \ => \ X\ will\ inform\ its\ neighbors$

c. $c(X,Y) = 3$

$=> \ D^X(A,Y) = 3 + 6 = 9 > 6 = D^X(A,W)$

$=> \ min(D^X(A,n))\ will\ not\ change$

$=> \ X\ will\ not\ inform\ its\ neighbors$

3.

D^X	Y	Z
Y	2	7+1=8
Z	2+1=3	7
D^Y	X	Z
X	2	1+3=4
Z	2+3=5	1
D^Z	X	Y
X	7	1+2=3
Y	7+2=9	1

No nodes will send updated values to their neighbors because no min_cost value is changed from the previous iteration

4.

a.

router	to	message
y	w	$D^y(x) = 4$
	z	$D^y(x) = 4$
z	y	$D^z(x) = 6$
	w	$D^z(x) = \infty$
w	y	$D^w(x) = \infty$
	z	$D^w(x) = 5$

b. Yes there will be a count-to-infinity problem.
$c(x,y): 4 \ => \ 60 \ between \ t0 \ and \ t1$

Time:	t0	t1	t2	t3
Router:				
y to w	$D^y(x) = 4$	$D^y(x) = \infty$		
y to z	$D^y(x) = 4$	$D^y(x) = 9$		
z to y	$D^z(x) = \infty$		$D^z(x) = \infty$	
z to w	$D^z(x) = 5$		$D^z(x) = 10$	
w to y	$D^w(x) = 6$			$D^w(x) = 11$
w to z	$D^w(x) = \infty$			$D^w(x) = \infty$

The shortest path from z to x is 50. Z will stop updating when the cost reaches 50. The value of z is incremented by 5 every 3 iterations. Therefore, it will take up to 30 iterations for the cost to reach 50. Since the starting value is 5, it may take 27-30 iterations.

c. Increase the cost of the y <-> z path to ∞. In other words, cut the path. The count-to-infinity problem occurs at $D^z(x,y)$

5.

a.

Prefix	Link	Notation
11111110	0	254/8
11111111 0000 0000	1	255.0/16
11111111	2	255/8
otherwise	3	otherwise

b. First: match to 11111111 00000000 entry -> link interface 1
   Second: match to 11111111 entry -> link interface 2
   Third: doesn't match to link interface 0, 1, 2 -> link interface 3

6.

2	4	8	16	32	64	128
1	2	3	4	5	6	7

Subnet 1: 223.1.17.0/25 7 digits left => 128 interface
Subnet 2: 223.1.17.128/26 6 digits left => 64 interface
Subnet 3: 223.1.17.192/26 6 digits left => 64 interface

7.

a. Subnet A: 152.83.254/24 8 digits left => 256 interfaces > 250
10011000 01010011 11111110

Subnet B: 152.83.255.0/25 7 digits left => 128 interfaces > 120
10011000 01010011 11111111 0

Subnet C: 152.83.255.128/29 – 152.83.255.255/29 128-8=120 interfaces
10011000 01010011 11111111 1

Subnet D: 152.83.255.248/30 2 digits left => 4 interfaces > 2
10011000 01010011 11111111 111110

Subnet E: 152.83.255.252/31 1 digit left => 2 interfaces
10011000 01010011 11111111 1111110

Subnet F: 152.83.254.255/31 1 digit left => 2 interfaces
10011000 01010011 11111111 1111111

b.

Router	Prefix	Link Interface
AFD	10011000 01010011 11111110	A
	10011000 01010011 11111111 1111111	F
	10011000 01010011 11111111 111110	D
CFE:	10011000 01010011 11111111 1	C
	10011000 01010011 11111111 1111111	F
	10011000 01010011 11111111 1111110	E
DBE:	10011000 01010011 11111111 111110	D
	10011000 01010011 11111111 0	B
	10011000 01010011 11111111 1111110	E

8. IP header = 20 bytes
   Original Datagram:
   - Length = 2400 (header (20 bytes) + message (2380 bytes))
   - Id = 422
   - Fragment = 0
   - Offset = 0

   Fragment 1:
   - Length = 700 bytes
   - Id = 422
   - Fragment = 1
   - Offset = 0

   Fragment 2:
   - Length = 700 bytes
   - Id = 422
   - Fragment = 1
   - Offset = 85

   Fragment 3:
   - Length = 700 bytes
   - Id = 422
   - Fragment = 1
   - Offset = 170

   Fragment 4:
   - Length = 2380 – (680*3) = 340 bytes
   - Id = 422
   - Fragment = 1
   - Offset = 255

   Total fragments = floor((2400-20) / (700-20)) = 4

9. Yes
   AS X has an agreement of peering with AS Y
   AS Y has an agreement of peering with AS Z
   - The BGP route trailers are held by each autonomous system
   - AS X doesn't know that AS Y has a path to AS Z
   - AS X never forwards traffic
   - AS Y should communicate to AS X that it has no path to Z
   - AS Z can transfer all of Y's traffic

   Therefore yes, BGP alone can accomplish the task.

10.

a.

Position=0	Length=234	Application data=214

b. If one or more segments are lost or erroneous the whole datagram is discarded. However, TSP will automatically invoke fast retransmission.