# HW5 Solutions

### Problem 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 |  | 6,x | 6,x | 8,x |
| 2 | xvu | 7,v |  |  | 6,x | 6,x | 8,x |
| 3 | xvuw | 7,v |  |  |  | 6,x | 8,x |
| 4 | xvuwy | 7,v |  |  |  |  | 8,x |
| 5 | xvuwyt |  |  |  |  |  | 8,x |
| 6 | xvuwytz |  |  |  |  |  |  |

### Problem 6

The question needs two assumptions: (i) it asks about “the number of iterations from when the algorithm is run for the first time” (that is, assuming the only information the nodes initially have is the cost to their nearest neighbors), (ii) there is no cycle/loop in the network with negative cost. We assume that the algorithm runs synchronously (that is, in one step, all nodes compute their distance tables at the same time and then exchange tables). We first prove the following lemma.

Lemma: After t iterations, each node knows the minimum-cost path of up to t+1 hops to other nodes at up to t+1 hops away.

Proof:

1. At t = 0, each node knows the cost of one-hop paths to its immediate neighbors by assumption.
2. Assume that the statement holds for t-1 iterations (t > 0), i.e., the distance variable Dv(t-1)(y) equals the minimum cost from v to y involving up to t hops. After the t-th iteration, the variable is updated according to the Bellman-Ford equation to:

where the “min” is taken over v in one-hop neighborhood of x. If there exists a neighbor v with finite then is the minimum cost from x to y involving up to t+1 hops. Otherwise (i.e., x is more than t+1 hops away from y), ⎕

Let  be the length of the longest path without loops between any two nodes in the network. Using the above lemma, after  iterations, all nodes will know the minimum-cost path of  or fewer hops to all the other nodes (as all nodes are at most d hops away). Since any path with greater than  hops will have loops and thus a greater cost than the path with the loops removed, the algorithm will converge in at most  iterations.

ASIDE: if the DV algorithm is run as a result of a change in link costs, there is no a priori bound on the number of iterations required until convergence unless one also specifies a bound on link costs.

### Problem 9

NO, this is because that decreasing link cost won’t cause a loop (caused by the next-hop relation of between two nodes of that link). Connecting two nodes with a link is equivalent to decreasing the link weight from infinite to the finite weight.

### Problem 11



|  |  |
| --- | --- |
|  |  |
| Router z | Informs w, Dz(x)=∞ |
|  | Informs y, Dz(x)=6 |
| Router w | Informs y, Dw(x)=∞ |
|  | Informs z, Dw(x)=5 |
| Router y | Informs w, Dy(x)=4 |
|  | Informs z, Dy(x)=4 |

1. Yes, there will be a count-to-infinity problem. The following table shows the routing converging process. Assume that at time t0, link cost change happens. At time t1, y updates its distance vector and informs neighbors w and z. In the following table, “🡪” stands for “informs”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| time | t0 | t1 | t2 | t3 | t4 |
| Z | 🡪 w, Dz(x)=∞ |  | No change | 🡪 w, Dz(x)=∞ |  |
|  | 🡪 y, Dz(x)=6 |  |  | 🡪 y, Dz(x)=11 |  |
| W | 🡪 y, Dw(x)=∞ |  | 🡪 y, Dw(x)=∞ |  | No change |
|  | 🡪 z, Dw(x)=5 |  | 🡪 z, Dw(x)=10 |  |  |
| Y | 🡪 w, Dy(x)=4 | 🡪 w, Dy(x)=9 |  | No change | 🡪 w, Dy(x)=14 |
|  | 🡪 z, Dy(x)=4 | 🡪 z, Dy(x)= ∞ |  |  | 🡪 z, Dy(x)= ∞ |

We see that w, y, z form a loop in their computation of the costs to router x. If we continue the iterations shown in the above table, then we will see that, at t27, z detects that its least cost to x is 50, via its direct link with x. At t29, w learns its least cost to x is 51 via z. At t30, y updates its least cost to x to be 52 (via w). Finally, at time t31, all nodes receive the updated distance vector from their neighbors and there is no more updating, and the DV algorithm has converged.

Note: The shortest paths and the distance vectors have converged at t30, but it takes one more iteration to receive the updated distance vectors from neighbors.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| time | t27 | t28 | t29 | t30 | t31 |
| Z | 🡪 w, Dz(x)=50 |  |  |  | via w, ∞ |
|  | 🡪 y, Dz(x)=50 |  |  |  | via y, 55  via z, 50 |
| W |  | 🡪 y, Dw(x)=∞ | 🡪 y, Dw(x)=51 |  | via w, ∞ |
|  |  | 🡪 z, Dw(x)=50 | 🡪 z, Dw(x)= ∞ |  | via y, ∞  via z, 51 |
| Y |  | 🡪 w, Dy(x)=53 |  | 🡪 w, Dy(x)= ∞ | via w, 52 |
|  |  | 🡪 z, Dy(x)= ∞ |  | 🡪 z, Dy(x)= 52 | via y, 60  via z, 53 |

1. cut the link between y and z. (Any c(y,z) > 54 will work.)

### Problem 14

1. eBGP
2. iBGP
3. eBGP
4. iBGP

### Problem 15

1. I1 because this interface begins the least cost path from 1d towards the gateway router 1c.
2. I2. Both routes have equal AS-PATH length but I2 begins the path that has the closest NEXT-HOP router.
3. I1. I1 begins the path that has the shortest AS-PATH.