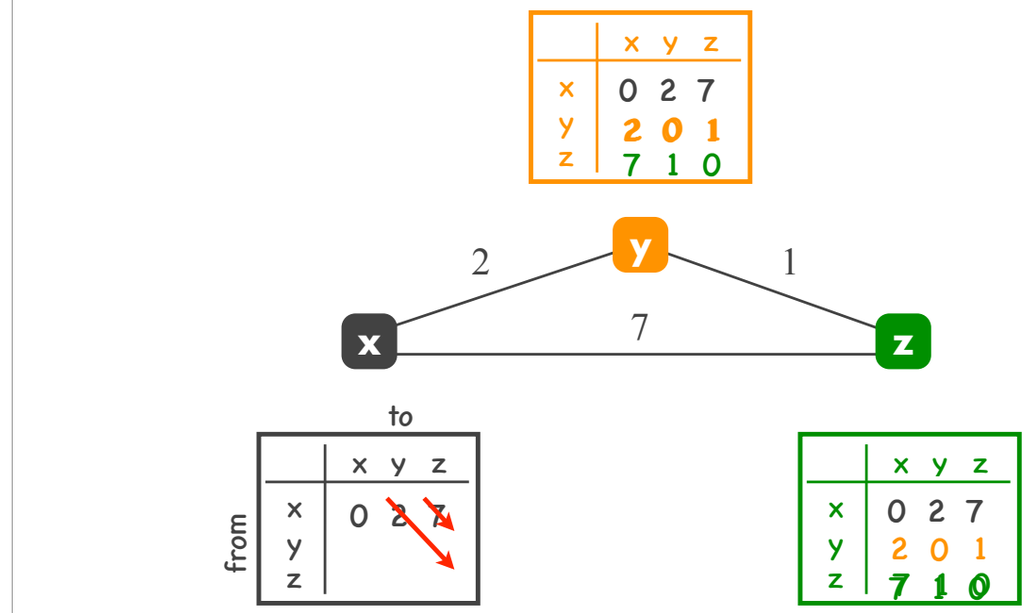


In the first round, all neighbors exchange tables.

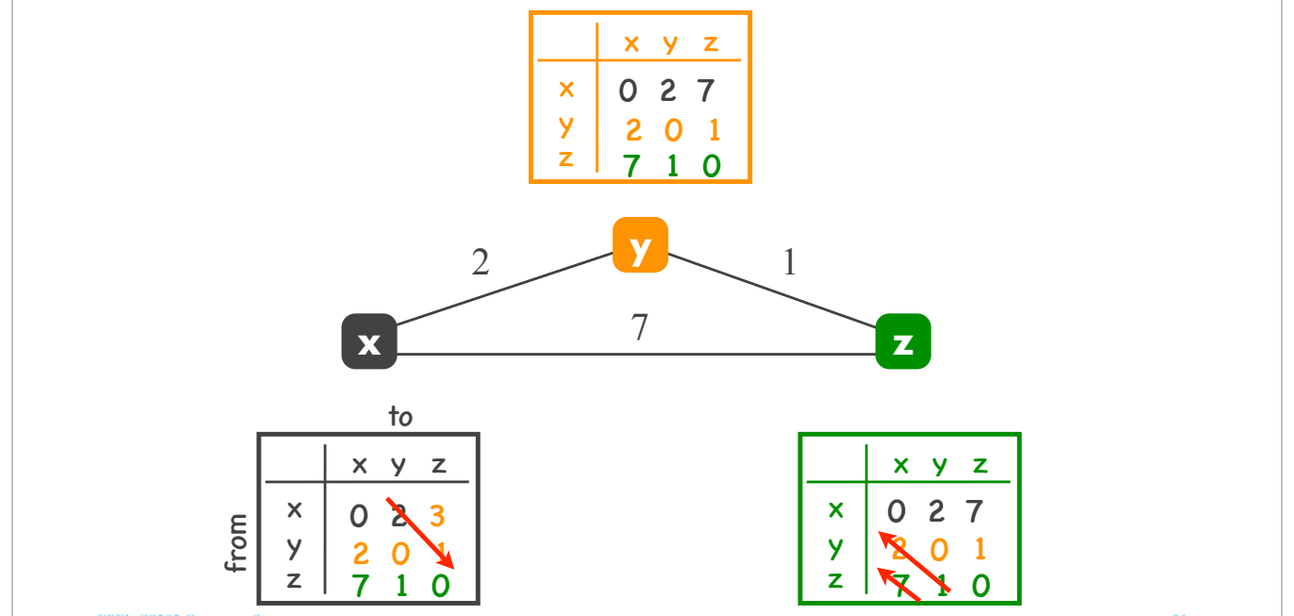
So, router x is going to learn some information from router y
and some other information from router z.

Similarly, routers y and z are going to learn new information.

Now, each router checks whether it can improve any of its current paths,
based on the new information it just learnt.

For example, concentrate on router x:

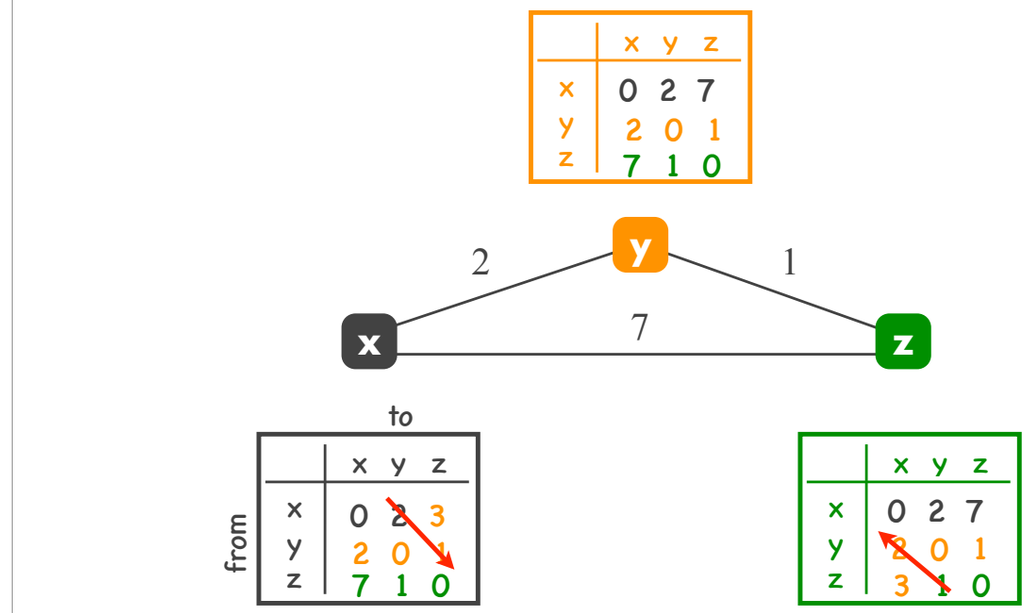
- Previously, router x could reach router z with cost 7.
- But now router x has learned that router y can reach router z with cost 1.
- And router x can reach router y with cost 2.
- This means that router x can reach router z with cost 3, through router y.



Hence, router x updates its structure with the fact that it can reach router z with cost 3.
I have made the number 3 orange to indicate that router x can reach router z with this cost through router y (the orange router).

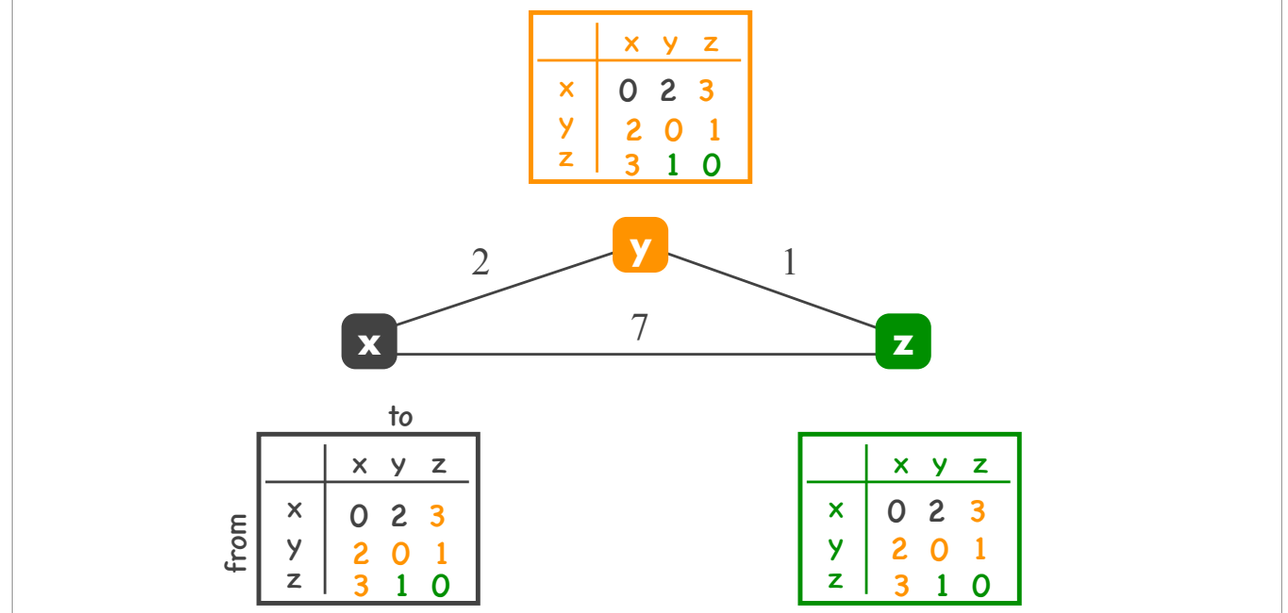
Now concentrate on router z:

- Previously, router z could reach router x with cost 7.
- But now router z has learned that router y can reach router x with cost 2.
- And router z can reach router y with cost 1.
- This means that router z can reach router x with cost 3, through router y.



Hence, router z updates its structure with the fact that it can reach router x with cost 3.
I have made the number 3 orange to indicate that router z can reach router x with this cost through router y.

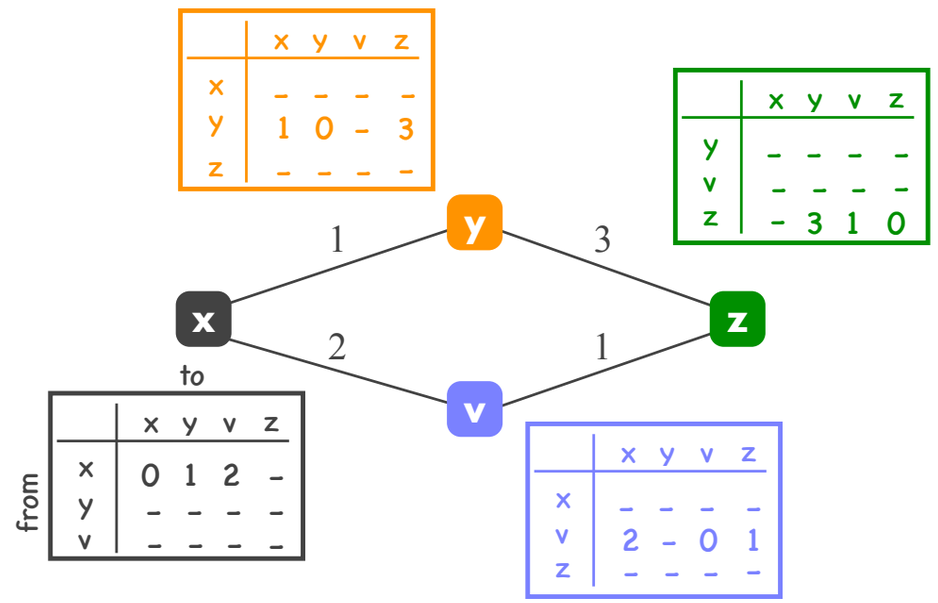
Finally, unlike routers x and z, router y cannot improve its paths with the new information it received. This is because router y's best paths to both x and z are direct paths that router y already knew about before exchanging any information with neighbors.



In the second round, all the neighbors exchange tables again.

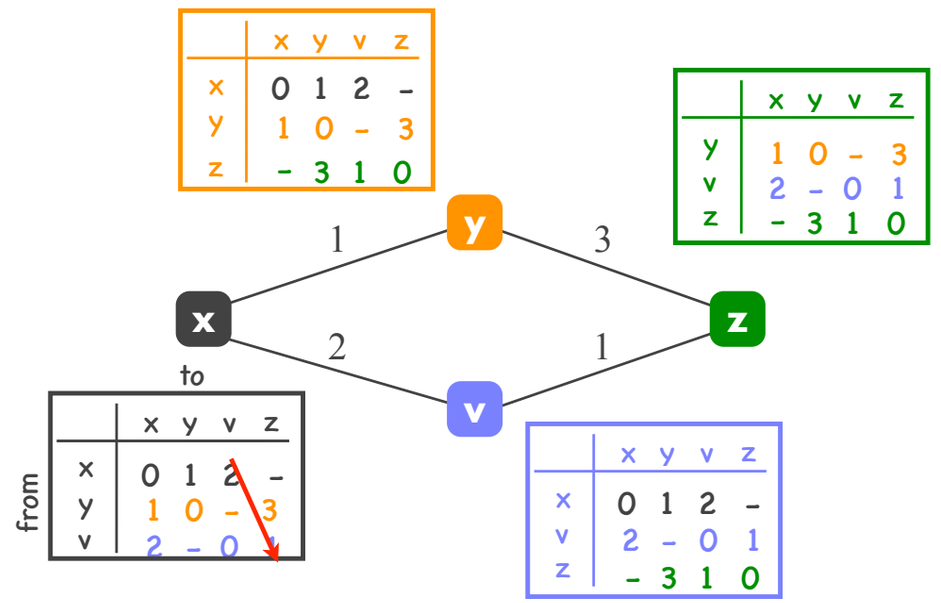
Each router checks whether it can improve its existing routes based on the new information it has just received.

In this example, they cannot, so the algorithm is done: each router has computed the least-cost path to every other router in the network.



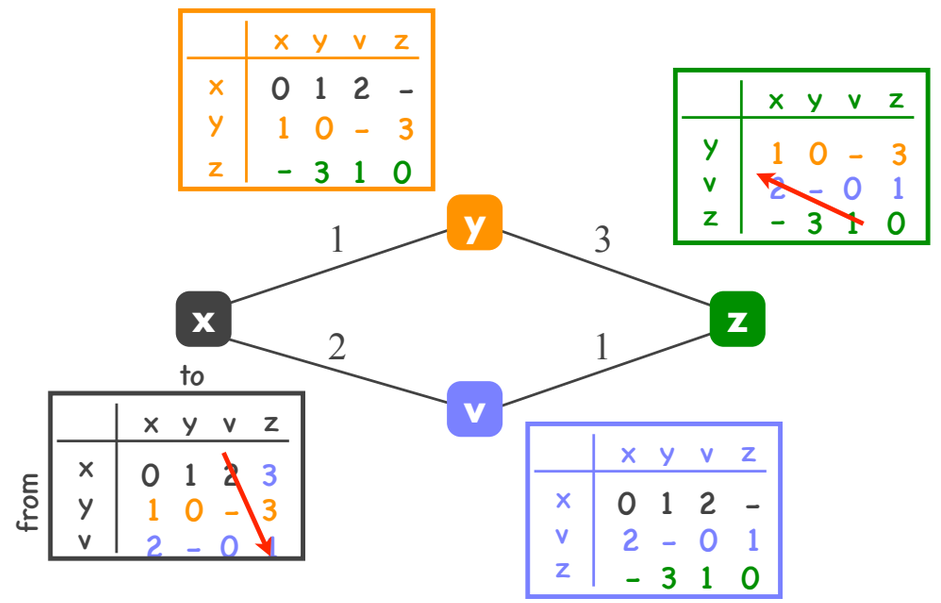
Here is a second, a bit more complicated example:

In the beginning, each router knows how to reach only its direct neighbors.



In the first round, all the neighbors exchange tables.

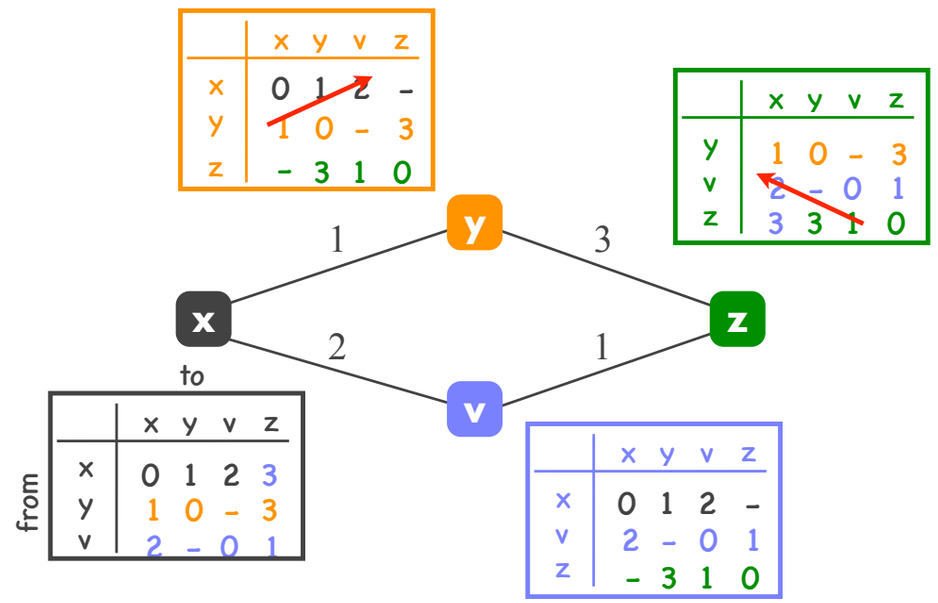
As a result, router x discovers a new path to router z, through router v, with cost 2+1=3…



and updates its table accordingly (note that the new number 3 that appears in x's table is purple, because it was computed based on information propagated through router v).

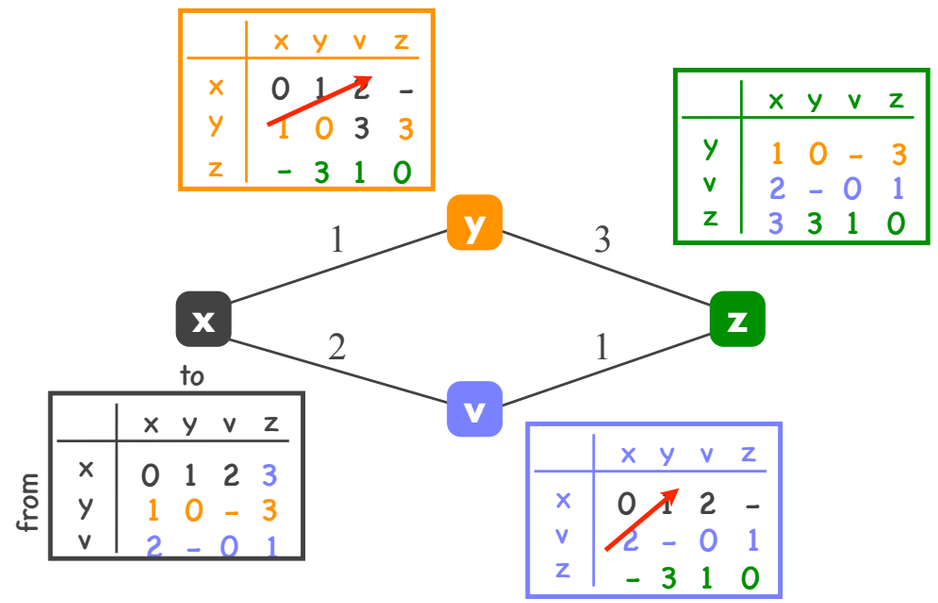
(Parenthesis: Router x actually discovers two new paths to router z: one through v, of cost 3, and one through y, of cost 4. It compares the two and determines that the one through v is better.)

Similarly, router z discovers a new path to router x, through router v, with cost 3…



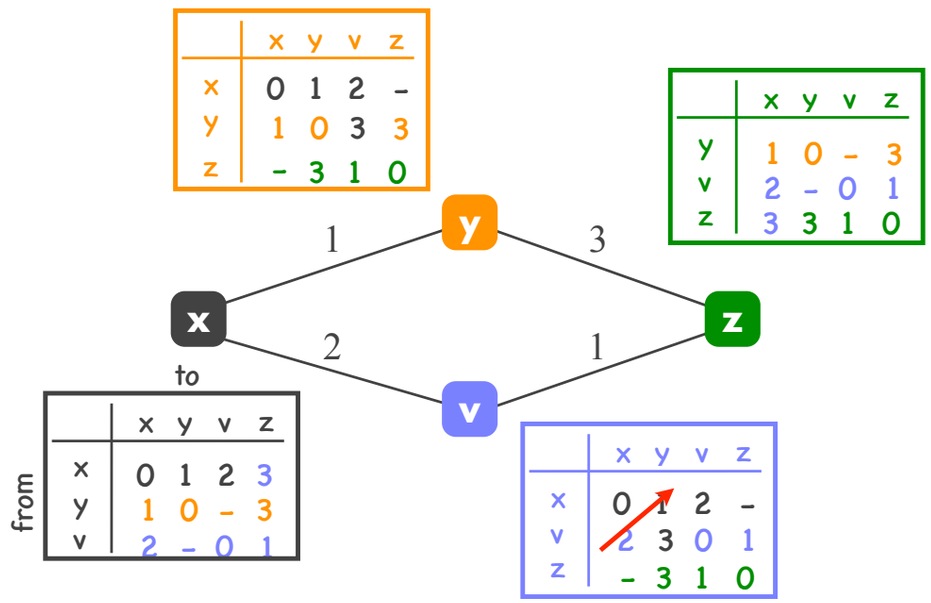
and updates its table accordingly.

At the same time, router y learns a new path to router v, through router x, with cost 1+2=3…

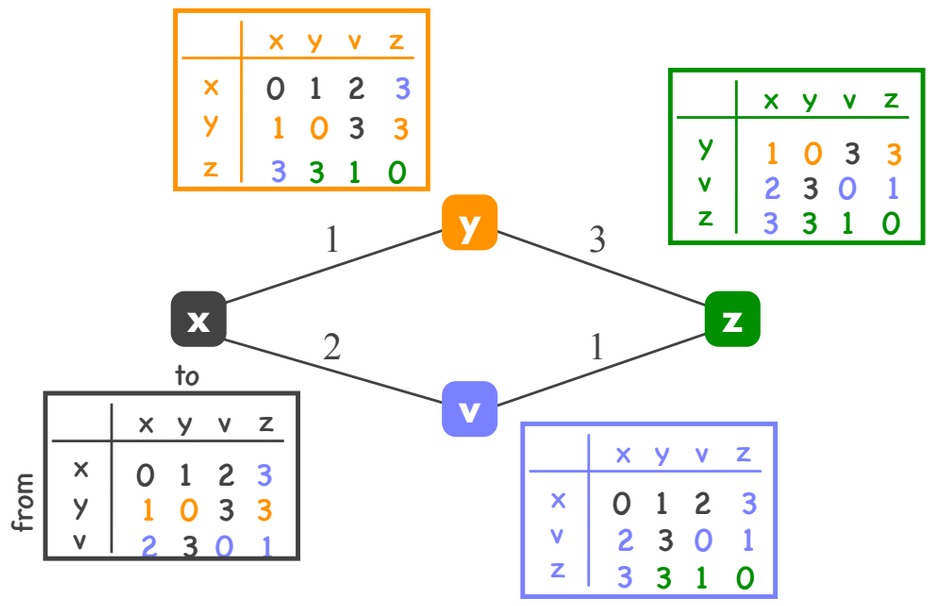


and updates its table accordingly (note that the new number 3 that appears in y's table is black, because it was computed based on information propagated by router x).

Similarly, router v learns a new path to router y, through router x, with cost 2+1=3…



and updates its table accordingly.



In the second round, all the neighbors exchange tables again.

This time, no router can improve any path with the new information that it has just received, so the algorithm is done.

Distance-vector routing algorithm

- Input to each router: local link costs
  & neighbor messages
- Output of each router: least-cost path
  to every other router

What we saw is an example of a distance-vector routing algorithm:

- In each round, the instance of the algorithm running at each router takes as input the router's current table plus the tables of its neighbors.
- At the end, the instance of the algorithm running at each router produces as output the least-cost path to every other router in the network.

Distance-vector routing algorithm

- "Distributed" algorithm
- All routers run it "together": neighbors exchange and react to each other's messages

A distance-vector routing algorithm is a "distributed" algorithm, in the sense that all routers run it "together": they exchange messages and they keep reacting to each other's messages.

Bellman-Ford algorithm

- All neighbors exchange information
- Each router checks whether it can improve current paths by leveraging the new information
- Ends when no improvement is possible

The particular distance-vector routing algorithm that we discussed is called the Bellman-Ford algorithm.

In summary:

- At each step, all neighbors exchange tables.
- Each router checks whether it can use the new information it just received to improve its current paths.
- The algorithm ends when no improvement is possible.

Link-state + distance-vector

- They solve the same problem:
  compute the least-cost path from each router
  router to each other router

We discussed two types of routing protocols: link state and distance vector.

They both solve the same problem: compute the least-cost paths between routers.

But, they do it…