**Warshall’s Algorithm**

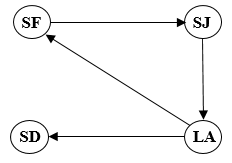
Dr. Byun

11/28/2020

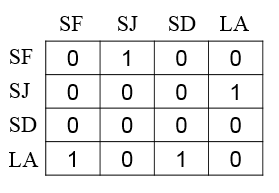
Warshall’s algorithm is an algorithm to compute the **transitive closure** of a directed graph, which means that we can identify the existence of nontrivial paths in a directed graph using the algorithm.

**Example**

Suppose we have a very small highway system connecting California's four cities (SF, SJ, SD, LA). As shown in the following graph, there are four direct connections from SF to SJ, from SJ to LA, from LA to SF, and from LA to SD in the highway system.

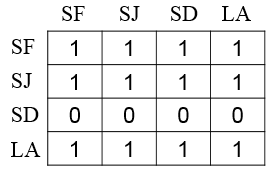


We can represent the cities and connections in the adjacency matrix like below:



In the matrix, the city name in the left column represents the source, and the city name in the top row represents the destination. Therefore, the value 1 in the first row and the second column indicates that there is a connection from SF to SJ. Similarly, the second row shows the connection from SJ to LA.

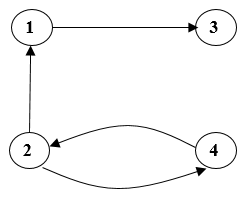
We now try to calculate the transitive closure of the matrix. The end result of the transitive closure is as follows.



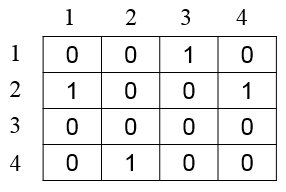
In the result of the first row, we know that there are paths from SF to SF, SF to SJ, SF to SF, and SF to LA. There is no direct connection from SF to LA in the initial matrix. However, we have confirmed that we can go from SF to LA through SJ. Of course, it is very simple to our eyes. But the question is how a computer program can find it out.

**Exercise**

Let’s assume that we have a directed graph with four nodes and four edges like below.



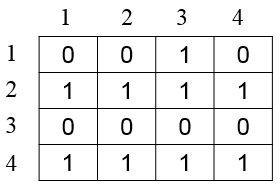
We can represent the graph using the adjacency matrix like below:



Determine the transitive closure of the graph. **Do not see the answer immediately.** Try to solve it by yourself.

**Solution**

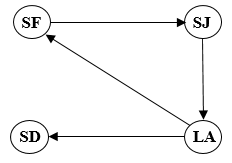
This is the transitive closure of the graph.



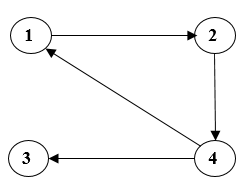
You can go all nodes from the node 2 and 4. However, you can go only to the node 3 from the node 1.

**Warshall’s Algorithm**

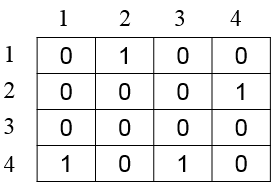
The **basic idea** of the Warshall algorithm is to **consider each node one by one**. Let's use the highway example again to make our discussion easier. This is the directed graph.



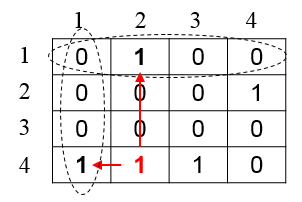
However, to easily implement the Warshall algorithm, let’s rename SF to Node 1, SJ to Node 2, SD to Node 3, and LA to Node 4. This is an updated direction graph.



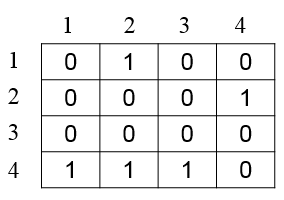
This is the adjacency matrix for the graph. At the Warshall’s algorithm, we call it **R(0)**.



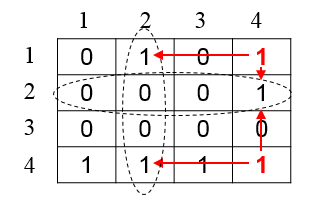
Now let's consider the node 1. In other words, pay attention to the first row and the first column of the matrix. We can see that there is a path from 4 to 1 and another path from 1 to 2 in the matrix. Thus, indirectly through the node 1, we can see a path from the node 4 to the node 2. We can explain the situation like this:



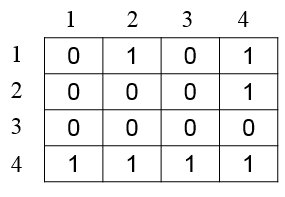
Note that the value of row 4 and column 2 become 1 because (4 → 1 AND 1 → 2) is (4 → 2). This is an updated matrix after considering node 1 and is called **R(1)**.



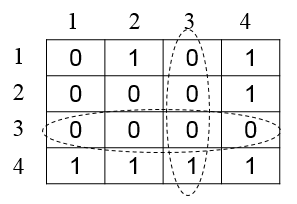
Now let's consider the node 2. In other words, pay attention to the second row and the second column of the matrix. We can see that there is a path from 1 to 2 and another path from 2 to 4. Thus, indirectly through the node 2, we know that there is a path from the node 1 to the node 4. (1 → 2 AND 2 → 4) makes (1 → 4). Similarly, we can find another path from 4 to 4 via the node 2. We can explain the two situations like this:



This is an updated matrix after considering node 2 and is called **R(2)**.

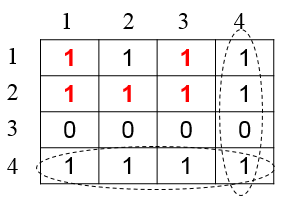


Now let's consider the node 3. However, this time there is no new path through the node 3.

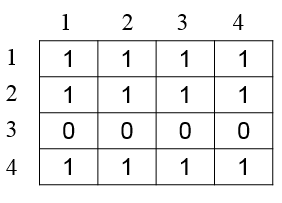


Thus, the new adjacency matrix (= **R(3)**) is the same as **R(2)**.

Finally, let’s consider the node 4. In the matrix we know that we can find five new paths:



This is the last adjacency matrix considering node 4 and is called **R(4)**.



In short, the Warshall’s algorithm calculates (n+1) matrices R(0), R(1), R(2), …, and R(n) where R(0) is the initial adjacency matrix and R(k)[i, j] = 1 if and only if there is a path from i to j with only first k vertices allowed as intermediate nodes.