**Graphs 1**

**Adjacency Matrix: Warshall's Algorithm**

Here is a little more interesting directed graph:

**4** Pierre

**0** Pendleton

**7** Pueblo

**6** Princeton

**3** Phoenix

**2** Peoria

**5** Pittsburgh

Here is the adjacency matrix for the graph above:

**1** Pensacola

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0.** | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | **7.** |
| **0. Pendleton** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| **1. Pensacola** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **2. Peoria** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| **3. Phoenix** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| **4. Pierre** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5. Pittsburgh** | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| **6. Princeton** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| **7. Pueblo** | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Notice that the numbering of the cities is arbitrary. Alphabetical order is only a convenience here.

**Assignment, part 1**

Add these user-friendly features to AdjMat, as specified in the Warshall interface.

* What data structures stores the names of the vertices and their vertex numbers?
* How will you read the names from a text file cities.txt?
* How will you display the names and their vertex numbers?
* How will you read the adjacency matrix from the text file citymatrix.txt ?
* Overload the isEdge method so that the user can deal with names as well as indexes.

One of the fundamental questions about graphs is the question of *reachability,* which we defined as the existence of a path from source to target. For example:

1. If you start from Phoenix, is Peoria reachable? \_\_
2. Is it possible to go from Peoria to Phoenix? \_\_
3. If you start from Peoria, is Princeton reachable? \_\_\_
4. Is it possible to go from Pensacola to Pendleton? \_\_\_
5. Is it possible to go from Pendleton to Pensacola? \_\_\_

Amazingly, Warshall's Algorithm solves the *all-pairs reachability* problem.

**Warshall's Algorithm**

Warshall's Algorithm starts with an adjacency matrix, examines the edges between three vertices, and adds edge information to the matrix. The completed matrix is called a *reachability* matrix. Here is Warshall's algorithm: for any given vertex v, if the paths [r, v] and [v, c] exist, then a path to [r, c] exists. Repeat for every vertex v.

What is the Big-O for Warshall’s Algorithm? \_\_**\_\_\_\_**\_\_ The disadvantage of Warshall’s Algorithm is that it may take too long to run in practice.

**Assignment, part 2**

In your AdjMat implement the Warshall interface. Implement Warshall's algorithm in allPairsReachability. You will turn in the improved AdjMat.

The WarshallDriver.java driver program, which is given to you, prompts the user for the necessary files (cities.txt and citymatrix.txt). Here is the output:

**Sample Run**

|  |  |
| --- | --- |
| Warshall's Algorithm! Enter file of names: cities Enter file of the matrix: citymatrix Adjacency Matrix 0 0 0 0 0 0 0 1  0 0 0 1 0 0 0 0  0 0 0 0 0 1 0 1  0 0 0 0 0 1 0 1  1 0 0 0 0 0 0 0  0 1 0 1 0 0 0 0  0 0 0 0 0 1 1 0  1 0 0 0 1 0 0 0  Number of Edges: 13  0-Pendleton 1-Pensacola 2-Peoria 3-Phoenix 4-Pierre 5-Pittsburgh 6-Princeton 7-Pueblo  Reachability Matrix 1 0 0 0 1 0 0 1  1 1 0 1 1 1 0 1  1 1 0 1 1 1 0 1  1 1 0 1 1 1 0 1  1 0 0 0 1 0 0 1  1 1 0 1 1 1 0 1  1 1 0 1 1 1 1 1  1 0 0 0 1 0 0 1  Number of Edges: 40 | Is it reachable? Enter name of start city (-1 to exit): Peoria  Enter name of end city: Phoenix true  Is it reachable? Enter start city (-1 to exit): -1  ================== EXTENSION ================== List of every city's reachable cities:  Pendleton--> [Pendleton, Pierre, Pueblo] Pensacola--> [Pendleton, Pensacola, Phoenix, Pierre, Pittsburgh, Pueblo] Peoria--> [Pendleton, Pensacola, Phoenix, Pierre, Pittsburgh, Pueblo] Phoenix--> [Pendleton, Pensacola, Phoenix, Pierre, Pittsburgh, Pueblo] Pierre--> [Pendleton, Pierre, Pueblo] Pittsburgh--> [Pendleton, Pensacola, Phoenix, Pierre, Pittsburgh, Pueblo] Princeton--> [Pendleton, Pensacola, Phoenix, Pierre, Pittsburgh, Princeton, Pueblo] Pueblo--> [Pendleton, Pierre, Pueblo]  List the reachable cities from: Pittsburgh [Pendleton, Pensacola, Phoenix, Pierre, Pittsburgh, Pueblo]  List the reachable cities from: -1 |

**Extension**

Write the getReachables method which accepts the name of a city and returns a list of that city’s reachable cities. This is harder than it seems. I had to add another data structure as a third field in AdjMat.