**Graphs 5**

**AdjList with Cities**

Pendleton Pueblo

Pensacola Phoenix

Peoria Pittsburgh

Peoria Pueblo

Phoenix Pittsburgh

Phoenix Pueblo

Pierre Pendleton

Pittsburgh Pensacola

Pittsburgh Phoenix

Princeton Pittsburgh

Princeton Princeton

Pueblo Pendleton

Pueblo Pierre

In the previous AdjList labs we hard-coded the vertices and edges in the driver. It would be nicer to read that data from a text file. First we will read the cities. Then we read the edges between the cities, in the format as shown at the right.

The graph is familiar:

**5** Pittsburgh

**6** Princeton

**1** Pensacola

**2** Peoria

**7** Pueblo

**3** Phoenix

**4** Pierre

**0** Pendleton

How many vertices are in this graph?\_\_\_\_

How many edges are in this graph?\_\_\_\_

Is every vertex reachable? Y/N If yes, then the graph is *strongly* *connected*.

If you recall, depthFirstSearch returned a List of reachable vertices from a given source.

Write the list of reachable cities from Pendleton:

Write the list of reachable cities from Pittsburgh:

**isReachable**

Assuming that you have a working depthFirstSearch, let’s use its results to answer specific city-to-city reachability questions. Write isReachable :

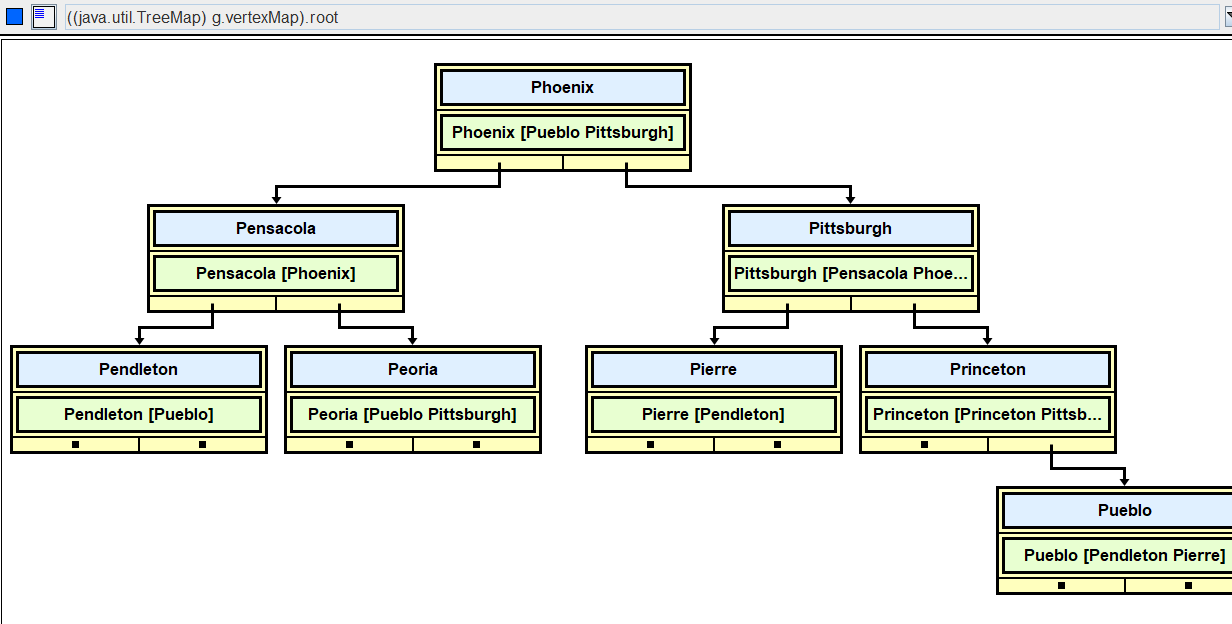
public boolean isReachable(String source, String target)  
 {

**Assignment**

In AdjList comment in and implement the EdgeListWithCities interface.

interface EdgeListWithCities  
{  
 public void readData(String cities, String edges);  
 public int edgeCount();  
 public int vertexCount();   
 public boolean isReachable(String source, String target);  
   
 public boolean isStronglyConnected(); //return true if every vertex is   
 //reachable from every other vertex, otherwise false   
}

Remember that we are still using a TreeMap:



Write the three methods below.

public int edgeCount()

{

}

public int vertexCount()

{

}

//A directed graph is "[strongly connected](https://mathworld.wolfram.com/StronglyConnectedDigraph.html)" if there is a directed  
// path between every pair of vertices.

//@return true if every vertex is reachable from every   
// other vertex, otherwise false

public boolean isStronglyConnected()

{

}

**Sample Runs** (AdjList\_5\_Driver with improved AdjList)

|  |  |
| --- | --- |
| Edge List with Cities!  Enter file of cities: cities Enter file of edges: cityEdges  The cities with their edges: Pendleton [Pueblo] Pensacola [Phoenix] Peoria [Pueblo Pittsburgh] Phoenix [Pueblo Pittsburgh] Pierre [Pendleton] Pittsburgh [Pensacola Phoenix] Princeton [Princeton Pittsburgh] Pueblo [Pendleton Pierre]  Number of vertices: 8 Number of edges: 13  Is this graph strongly connected? false  Can you get there from here?   Start at (-1 to exit): Pittsburgh  End at: Pueblo  true  Can you get there from here?   Start at (-1 to exit): Pueblo  End at: Pittsburgh  false  Can you get there from here?   Start at (-1 to exit): -1 | Edge List with Cities!  Enter file of cities: threeCities Enter file of edges: threeCitiesEdges  The cities with their edges: Pendleton [Pueblo] Pierre [Pendleton] Pueblo [Pendleton Pierre]  Number of vertices: 3 Number of edges: 4  Is this graph strongly connected? true  Can you get there from here?   Start at (-1 to exit): Pendleton  End at: Pierre  true  Can you get there from here?   Start at (-1 to exit): -1 |