CS 445

Rec 8

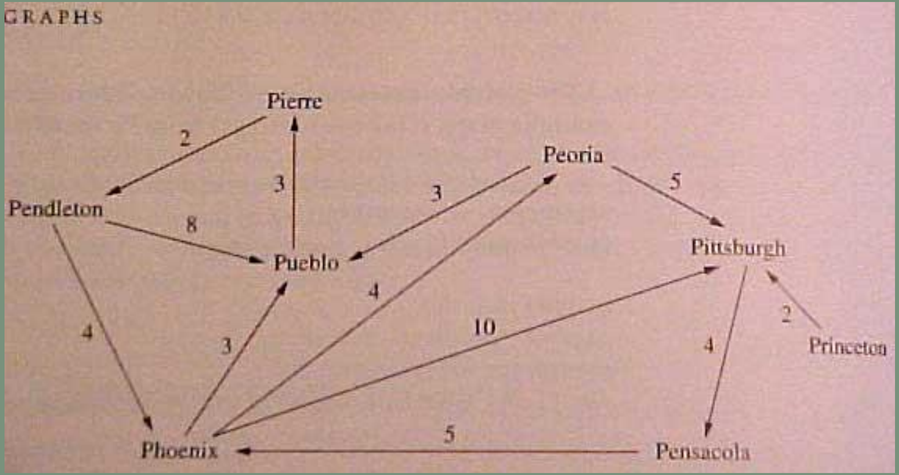
Agenda

1. **Topic of This Week**
   a. Implementing a Graph with 2D Array
   b. Hands-on practice on 2D array manipulation through Lab 7
2. **Working Session: Lab 7**

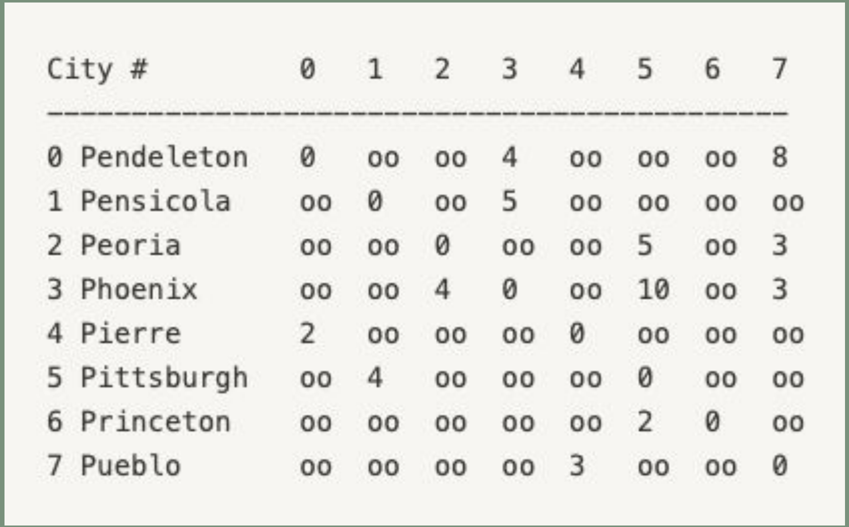
Lab 7

- Create a Graph to represent this mini world
  - Use a 2D array as the underlying data structure
  - Calculate the connectivity of the cities

- A graph may be thought of as a set of nodes and edges
  - Since this is a weighted, directed graph, the 2D array is thus structured with each element being G[src][dest] = weight



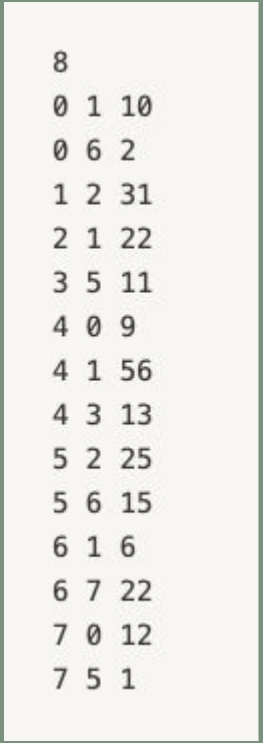
Adjacency Matrix as Data Structure



```
City #          0   1   2   3   4   5   6   7
---------------------------------------------
0 Pendeleton    0   oo  oo  4   oo  oo  oo  8
1 Pensicola     oo  0   oo  5   oo  oo  oo  oo
2 Peoria        oo  oo  0   oo  oo  5   oo  3
3 Phoenix       oo  oo  4   0   oo  10  oo  3
4 Pierre        2   oo  oo  oo  0   oo  oo  oo
5 Pittsburgh    oo  4   oo  oo  oo  0   oo  oo
6 Princeton     oo  oo  oo  oo  oo  2   0   oo
7 Pueblo        oo  oo  oo  oo  3   oo  oo  0
```

- This is what the data structure look like to represent our graph
  - Notice that the weights on the main diagonal line is 0 because the distance from one city to itself is 0
  - If infinity means there's no edge existed between two cities (we use -1 to denote NO_EDGE in the code)
- Information stored in the 2D array
  - Row #: source city
  - Col #: destination city
  - G[row][col]: weight/distance

Adjacency List as Input File



```
8
0 1 10
0 6 2
1 2 31
2 1 22
3 5 11
4 0 9
4 1 56
4 3 13
5 2 25
5 6 15
6 1 6
6 7 22
7 0 12
7 5 1
```

- **This is the input file that we will use to create the 2D array**
  - **First line contains only one number representing the number of nodes in the graph**
- **Follows the format** source, dest, weight

Understanding Connectivity

Connectivity:

- **Out-degree**
  - The number of edges *leaving* the node
- **In-degree**
  - The number of edges *entering* the node
- **Degree**
  - The number of edges *entering and leaving* the node

What does it mean in a 2D array space?

- **In-Degree for Node 4:**
  - Go to the 4th column and calculate the number of valid distances
- **Out-Degree for Node 4:**
  - Go to the 4th row and calculate the number of valid distances
- **Degree:**
  - In-degree + out-degree

Customize Exception Error Messages

```
try {
    <code here>
    if ( <something goes wrong > )
        throw new Exception ( <" your customized exception error messages"> );
}
catch (Exception e) {
    System.out.println(e);
    System.exit(0);                    // optional: end the program if things go wrong
}
```

Your Tasks

- loadGraphFile()
  - Make use of addEdge()
- hasEdge()
- inDegree()
  - Make use of hasEdge()
- outDegree()
- degree()
- removeEdge()
- maxOutDegree()
- maxInDegree()
- minOutDegree()
- minInDegree()
- maxDegree()
- minDegree()