GROUP PROJECT FOR MSOR 492

Texas Airways has started service to Dallas, Houston, Austin, Corpus Christi, San Antonio, Lubbock, and El Paso. Fares between cities having a nonstop flight are given in the file *TexasAirways.xlsx*. The Vice President of Operations has asked your team to develop a program that displays the least-cost sequence of flights connecting any starting city to any ending city — for example, Dallas and El Paso. Your program should do this for as many starting and ending cities as the user wants and should work if the data in the Excel file change (including the number and names of the cities).

This project requires writing a Python program that includes the following two parts:

- For a network having *n* nodes and some arcs *ij*, each of which has a nonnegative distance d_{ij}, create an algorithm to compute two different (n x n) matrices $n + 1$ times, the last of which contains the lengths of the shortest paths between all pairs of nodes and the ability to create the associated shortest paths, as follows:

$$D_{ij}^{(0)} = \begin{cases} 0 & \text{if } i = j \\ \infty & \text{if } ij \text{ is not an arc} \\ d_{ij} & \text{if } ij \text{ is an arc} \end{cases} \qquad P_{ij}^{(0)} = \begin{cases} i & \text{if } i = j \\ -1 & \text{if } ij \text{ is not an arc} \\ j & \text{if } ij \text{ is an arc} \end{cases}$$

$$D_{ij}^{(k)} = \begin{cases} D_{ij}^{(k-1)} & \text{if } D_{ij}^{(k-1)} \le D_{ik}^{(k-1)} + D_{kj}^{(k-1)} \\ D_{ik}^{(k-1)} + D_{kj}^{(k-1)} & \text{otherwise} \end{cases} \qquad k = 1, \ldots, n$$

$$P_{ij}^{(k)} = \begin{cases} P_{ij}^{(k-1)} & \text{if } D_{ij}^{(k-1)} \le D_{ik}^{(k-1)} + D_{kj}^{(k-1)} \\ P_{ik}^{(k-1)} & \text{otherwise} \end{cases} \qquad k = 1, \ldots, n$$

  Note: Your program should work for any data in an Excel file that is in the same format as *TexasAirways.xlsx*, with the city names in the first row and first column and the distances always starting in cell B2.

- Allow the user to enter starting and ending nodes as many times as they want and, for each such pair, say *i* and *j*, compute and display both the length of the shortest path from *i* to *j* as well as the shortest path itself, which you can create by using the last matrix $P^{(n)}$, obtained from above, as follows:

```
$path = [i]$
while $i \neq j$
        $i = [P^{(n)}]_{ij}$
        $path = path + [i]$
```

In addition, each group should:

- Prepare a PowerPoint ***professional business presentation for the Vice President of Operations*** of **not more than 15 minutes** that includes, amongst other things, a demonstration of your program. You can decide which student talks about which topics, but ***each student must present for at least 4 minutes.***

- Prepare a brief User Report in a Word file that describes what you have done and how the user would use your program to solve this problem in general. Illustrate this with the data in the file *TexasAirways.xlsx*.

- ***On or before the due date in the syllabus, e-mail ME your Jupyter Notebook file, your Power Point presentation file, and your Word file with the User's Manual THAT ALL MUST WORK CORRECTLY ON A PC. The names of all your files should be ProjectGroup.ipynb*** followed by the number of your group (for example, *ProjectGroup5.ipynb*, *ProjectGroup5.ppt*, and *ProjectGroup5.doc*).

Your will be graded on the following aspects:

- Ease of use for the user, which includes the degree to which the process is automated and requires less for the user to do.
- Quality of the displayed data and solution.
- Quality of your macros.
- Quality of your User Report.
- Quality of your presentation.
- Following the given instructions.