

Assignment Problem Algorithm Assignment

Design and write an algorithm in Python for solving the Transportation/Assignment Problem.

The programming to retrieve the data from the database has already been done for you, as has the computation of distances between all pairs of location. This permits your focus to be on constructing the MST algorithm.

Develop your algorithm within the provided Python program by doing the following:

- Retrieve the assignment materials from Github repository:
 - `TransDev.py`
 - `assign.sql` (MySQL dump file)
 - Github URL: <https://github.com/jrb28/BUAD5042Trans>
- Implement your algorithm in the Python program named `TransDev.py` in the function named `trans()`
 - Be sure to change the MySQL connection parameters at the top of `TransDev.py` as is appropriate for your MySQL instance in order to connect to the database
- Use `TransDev.py` to test your algorithm for the problems in the accompanying MySQL database.
 - A link to download a dump file for the `assign` database is located on the Github site mentioned above, which contains multiple problems.
- Once you have completed developing your algorithm, then cut and paste the `trans()` function and save it in a file entitled `transportation.py`
- Submit your assignment by placing the file `transportation.py` in the folder in the location below. Replace `[username]` with your username.
 - `\\jonesfiles.campus.wm.edu\acstore-classes\BUAD5012\student\[username]`
- Algorithm/Function Input parameters: these variables have already been populated vis-à-vis the statements in the main program of `TransDev.py`.
 - `dist`: a dictionary where the key is a tuple indicating a combination of one store ID and one DC ID in the form `(dc_id, store_id)` and the value is the distance between the two locations.
 - `dcs`: a dictionary of the DCs where the key is a unique integer that identifies each DC and the value is a 3-tuple indicating the DC constraints in the form of `(max daily cubic feet vol., num. of doors, num. of drivers)`
 - `stores_vol`: a list of lists where each sub-list is of the form `[stores_id, daily cubic feet of goods needed]`
- Algorithm/Function Output parameters: A return statement has been provided in the `trans()` function that returns the two required parameters
 - `name_or_team`: assign a string value to this variable with either your W&M username or your first and last name
 - `result`: a Python list of tuples containing your solution. Each tuple in the list is of the form `(store_id, dc_id)` where the `store_id` values correspond to

the data in the `stores_vol` dictionary and the 0-th element of the dictionary keys in `dist`, and the `dc_id` values are the keys in the `dcs` dictionary. For example, the `result` list,

$$[(0,1),(1,2),(2,1),\dots]$$

indicates that the stores with `store_id` 0 and 2 have been assigned to the DC with `dc_id` 1 and `store_id` 1 has been assigned to `dc_id` 2. Each `store_id` can be listed only once and no DC can be overloaded along any one of the constraint dimensions, processing volume, number of truck doors, or number of available drivers.