Minimal Spanning Tree (MST) Algorithm Assignment

Design and write an algorithm in Python for solving the MST Problem. In this case, you will be developing an MST algorithm to connect Dollar Tree stores near Chesapeake, VA. This MST solution might be a starting point for a Traveling Salesperson Problem for these same locations.

Much of the programming to get the data out of the database has already been done for you as has the computation of distances between all pairs of location. This permits you to focus your efforts on constructing the MST algorithm. In that regard:

- The variable `locs` contains the data about each location taken from the database including location id, latitude, longitude, address, and type of facility (distribution center or store). The database contains 105 stores and one DC for Dollar Tree.
- The Haversine formula is used for the distance between all locations pairs. The provided function computes distances from the latitude and longitude values. Further, two nested loops are provided in the main program to compute the distance between all pairs of locations and that information is put into a dictionary called `dist`.

For future reference, this is a good reference for Haversine formulas for distance and bearing:

http://www.movable-type.co.uk/scripts/latlong.html

The `mst_algo()` function, in which you will code your MST algorithm takes the two data structures listed above as input parameters, `locs` and `dist`.

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

- Retrieve the assignment materials from Github repository:
  - `MSTDev.py`
  - `mst.sql` (MySQL dump file)
  - Github URL: https://github.com/jrb28/BUAD5042MST
- Implement your algorithm in the Python program named `MSTDev.py` in the function named `mst_algo()`
  - Be sure to change the MySQL connection parameters at the top of `MSTDev.py` as is appropriate for your MySQL instance in order to connect to the database
- Use `MSTDev.py` to test your algorithm for the problems in the accompanying MySQL database.
  - A link to download a dump file for the database `mst_dt` is located on the Github site mentioned above.
  - The `mst_dt` database contains two problems. One problem is small and you should be able to check manually whether you obtained the correct solution. This data is not Dollar Tree data. The second problem is Dollar Tree data and it is a larger problem.
- Once you have completed developing your algorithm, then cut and paste the `mst_algo()` function and save it in a file entitled `mst.py`
- Submit your assignment by placing the file `mst.py` in the folder in the location below. Replace [`username`] with your username.
  - `\\jonesfiles.campus.wm.edu\acstore-classes\BUAD5042\student\[username]`

- Output parameters: A return statement has been provided in the `mst_algo()` 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
  - `mst`: this variable is for your solution, which should be a list of tuples. The tuples indicate that links in your MST solution. For example, this list of tuples indicates that the connections in the MST solution as between locations 0 and 1, between 0 and 2 and between 2 and 3:

    [(0,1),(0,2),(2,3)]