ASSIGNMENT 6 - Travelling Salesman BY EE22B025

GOAL OF THE PROBLEM :

- The main goal of the problem is to find a *sequence of numbers* which specifies the order in which to visit the cities. This order of cities must be such that the distance travelled from one city to other should be **minimum**.

Explanation of Functions and Approach:

Unpack_Coord(cities) :

- The purpose of this function is that, given a list of tuples (which contains the x and y coordinates), it returns two numpy arrays containing the x and y coordinates seperately, thus *unpacking* the list of tuples.
- We need this function because we require x and y coordinates seperately so that we can rearrange them in the correct order of cities which is given by the ***tsp(cities)*** function.

distance(cities , cityorder) :

- This function basically takes in the sequence of city coordinates(*cities*) and the *cityorder* as well and then finds the distance it takes to traverse across the cities in the order specified.
- The function first gets the x and y coordinates and then rearranges these coordinates in the given *cityorder* and then returns the **total distance** including the distance of returning back to the first city.

tsp(cities) :

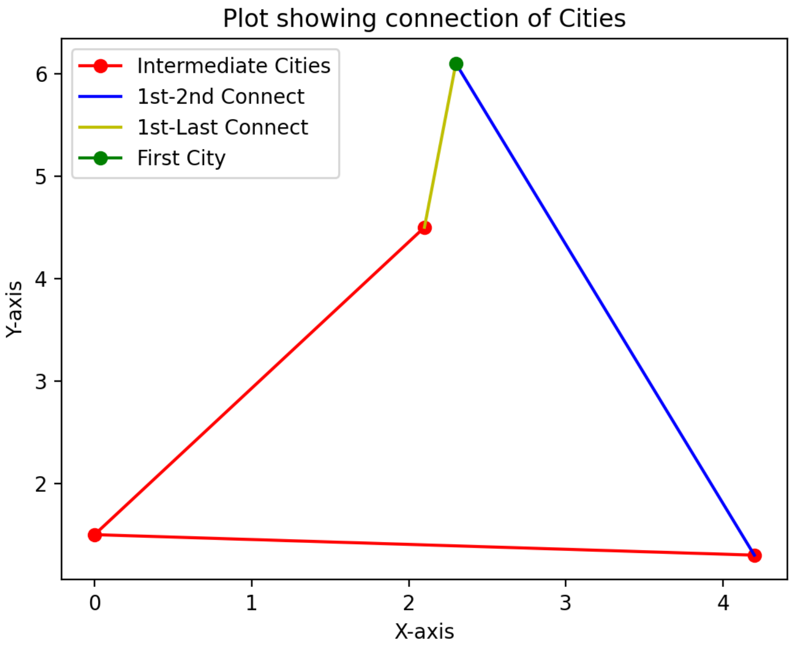
- This is the function where the ***Simulated Annealing Algorithm*** is used.
- The basic algorithm of Simlulated Annealing in context of **TSP** is explained below :
  1) First we define all the Parameters - T , Decay_Rate, initial_guess.
  2) The idea is that for every iteration, we *randomly change* the cityorder. But this change should be only a little random. In this case, I have randomly selected *two elements* from the cityorder and **interchanged them** to generate a new cityorder. Then I find the distance d corresponding to this cityorder.
  3) We then compare d with $InitialGuess$.
     - If $d < InitialGuess$ then we update Intial_Guess to d.
     - If $d > InitialGuess$ then we generate a random probabilty p in $[0, 1)$ and then compare that with $e^{-\frac{\Delta E}{kT}}$, which depends on the Parameters. Again, if $p < e^{-\frac{\Delta E}{kT}}$ then update Initial_Guess, else proceed as it is using `pass` command.
     - In our case, ΔE is the (*Current_Random_Distance* - *Previous_Guess*).
  4) We keep repeating this process until the **For Loop Limit** is reached and then return the final *cityorder*.

Result for the given test file (4 cities) :

- Upon running the code with the given 4 city coordinates, we get the following result :
  1) The City Order is *[2 1 3 0]* and the minimum distance is ***d = 14.64154124236167.***
  2) The first random distance guess generated was ***17.794818674426676.***
  3) The percentage improvement is ***17.7201998500645.***
- NOTE:
  - Even if we rotate the city_order array, we will get the same minimum distance.
  - Every time you run the code, the *first random distance* changes and thus *percentage improvement* also changes. Along with these two, the *city_order* and the *graph* may also change.
  - But the **minimum distance** remains the same.

PLOT of 4 city test case :

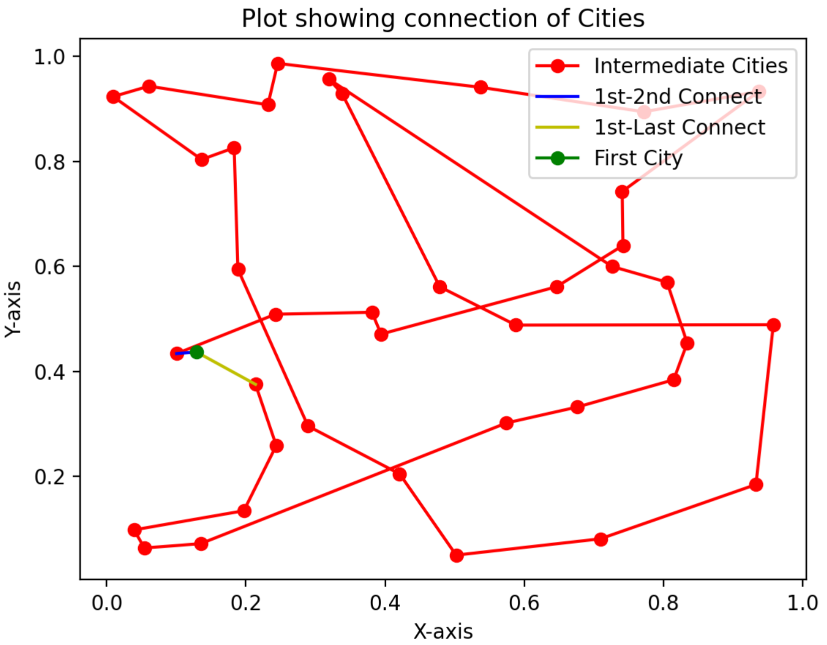
- **NOTE regarding Legend**:
  1) The RED dots represent the intermediate cities.
  2) The GREEN dot represents the First/Starting City.
  3) The BLUE line represents the connection between 1st and 2nd city.
  4) The YELLOW line represents the connection between 1st and Last city.



RESULT and PLOT of 40 city test case :

- Upon running the code with the given 4 city coordinates, we get the following result :
  1) The City Order is ***[15 23 25 16 1 31 8 35 2 7 32 5 29 0 9 28 37 39 11 17 19 20 30 33 6 36 4 12 27 13 24 18 26 21 38 10 22 34 3 14]*** and the minimum distance is $\boldsymbol{d} =$ ***7.155829730703569***
  2) The first random distance guess generated was ***20.365669083061096.***
  3) The percentage improvement is ***64.86327210012782.***

- The Plot is given below:



Instructions on how to run the Code :

- `numpy` and `matplotlib` libraries should be available.
- You have to make sure the test file should be in the same directory as the Python Notebook.
- You need change the filename to your "testfile name" in the code (where Reading_File() is called).
- Run ALL the cells at once.