Q1: A pilgrim or travelling salesman want to visit 50 locations in such a way that total travel time is minimum. We have distance matrix d(i, j) between 1 and 999 where i and j are any locations among 50 locations. This distance matrix is symmetric d(i, j) = d(j, i). As one can not go from same location i to i, we can use d(i,i) = 9999 to ensure algorithm does not select i to i route. We want you to implement travelling salesman problem algorithm as follows.

Step 1: Among all locations i, Connect a location (say i') to nearest neighbor j' if second penalty is highest among all locations.

Second penalty= Distance of location i to second nearest neighbor – Distance of location i to nearest neighbor

Step 2: d(i', j) = 9999 for all j. d(i, j') =9999 for all i. For any connected sequence if start location is i* and end location is j*, d(j*,i*)= 9999.

Step 3: Recalculate second penalty. Keep repeating step 1 & 2 till

either one of condition (i) & (ii) is satisfied for each location

(i) location has second penalty exceeding 999

(ii) location has minimum row entry as 9999.

At that point, connect everyone left to their nearest neighbor.

Return complete route & total distance by coding this algorithm in java, python, R or octave

Example for 4 location problem is attached as Q1Example.xls with this email. In same file, distance worksheet in excel file has distance matrix for 50 locations. As this matrix is symmetric, you can use code to create full distance matrix using d(i,j) = d(j,i).

Q2: We have distance matrix D(i,j) between every location i & j in the city. I=1 and I =2 represents 2 post office locations in Ooty. We have 100 customer locations in Ooty. 5 postmans are at postoffice 1 (postman 1 to 5) and other 5 at post office 2 (postman 6 to 10). A postman need to go from post office location to 10 customer locations every day to deliver letters and returns to whichever postoffice is closest to last customer location.

We will be sending 10 postmans in the city using nearest neighbor algorithm. We will first send postman 1 to nearest location to his postoffice. Then he will go to next closest location out of remaining 99. Then next closest out of remaining 98 …. After covering 10 locations, he will return to postoffice.

Next postman will chose nearest location out of remaining 90. Then go to next nearest out of 89…

(a) Write flowchart for this algorithms

(b) Write pseudo code or code (whichever is easier) for this problem for this algorithm