Map Anything Challenge Discussion

D'Azevedo, Gloria
gloria.dazevedo@gmail.com

May 2018

1 Question 1: Create an adjacency graph

1.1 Approach 1: List Solution

The idea of using this solution was such to preserve the order of the locations (only denoted by a latitude and longitude without a unique identifier) and then create an adjacency matrix for distances between points i and j. This solution when implemented correctly would save about half of the storage space to be on the magnitude of $n^2/2$ instead of n^2.Assuming the distance between two points are the same, no matter the direction, we only need to construct the upper right half of the matrix denoted $x_{ij}, j > i$. Also note that the diagonal would always be 0 since a point is always 0 distance from itself. I have implemented a "list of lists" solution where the wrapping list denotes the row number while the second index is a function of both the row number and the location number.

There are limitations to this approach. To access the distances between two points, the user would first have to know the order of the locations in the input file which would not necessarily be the easiest to go through. The current assumed implementation does not have a key or location identifier which could potentially make that lookup easier as well. In addition, due to the construction of the list of lists, the length of each sublist decreases as the rows increase. Thus, to access $A[i, j]$ as expected in mathematical notation, the actual lookup would look more like $A[max(i, j), min(i, j)] = A[i][(j - 1) - i]$ assuming that $i < j$

1.2 Approach 2: Dictionary Solution V1

1.3 Approach 3: Dictionary Solution V2

2 Question 2: Optimize the route

Goal: Need to minimize the number of drivers such that all clients are mostly satisfied, we can make all deliveries within the week and other constraints such as client service time and travel time are reasonable.