Assignment 4

Mehmet Kiraz 28375

Summary for the Algorithm

First I decided that the Floyd Warshall Algorithm was the best option for this question although it needed some upgrades and tweaks. In our question there are not one but 2 ways to go to another city. So to implement this via Floyd Warshall I doubled the main matrix size. The matrix had the distances for Bus stations and Train stations. The even indices had busses and the odd indices had trains. This way all the edges will be in one matrix therefore making the Floyd Warshall Algorithm applicable. After that I just get the wanted distance from the matrix without any problem.

Recursive Formulation

$$distanceMatrix^{K}(i, j) = 0 \quad \text{if } i = j$$

$$distanceMatrix^{K}(i, j) = distanceMatrix^{K}(i, j) \quad \text{if } k = 0$$

$$distanceMatrix^{K}(i, j) = \min[distanceMatrix^{K-1}(i, j), \quad \text{if } k \geq 1$$

$$distanceMatrix^{K-1}(i, k) + distanceMatrix^{K-1}(k, j)]$$

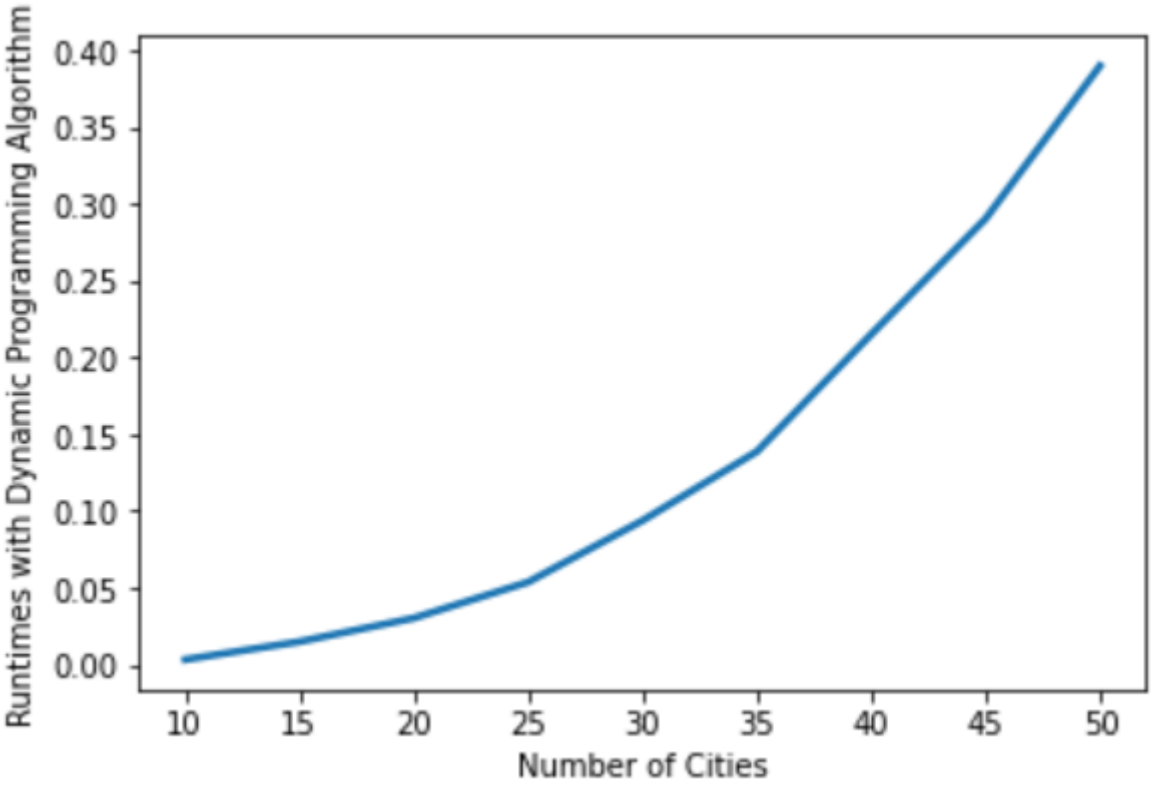
Pseudocode of the Algorithm

```
matrixLen = len(distanceMatrix[0])

for k in range(matrixLen):
    for i in range(matrixLen):
        for j in range(matrixLen):
            distanceMatrix[i][j] = min(distanceMatrix[i][j], distanceMatrix[i][k] +
            distanceMatrix[k][j])

if distanceMatrix[start][targetStationBus] ≤ distanceMatrix[start][targetStationTrain]
    return distanceMatrix[start][targetStationBus]
else
    return distanceMatrix[start][targetStationTrain]
```

Time Complexities

The main algorithm has 3 for loops which results in N^3 for the running time but the for the space time it needs is N^2 since we store the data in a two dimensional matrix.

Experimental Evaluations



The graph is definitely what I expected from my theoratical result. From the graph it can be seen that it is polynomial and support our theory.