**CSE2046 HOMEWORK 3 REPORT**

In this homework we are asked to implement an algorithm on ***Travelling Salesman Problem***to find a tour that total length approximates to optimum solution.

I first implemented direct nearest neighbor algorithm, then I use same algorithm by placing all cities as starting city and find the best solution among them.

**>>** python yasin\_enes\_polat.py inputfilename.txt **<<**

My program first reads input text file and creates City objects according to given informations, City object has x & y coordinates, id attributes and information that contains whether the city added to tour or not. Then program starts to create adjacency matrix, while filling adjacency matrix, my program creates Node object which has distance (i to j) and id (j) attributes in it, I use these objects to keep corresponding IDs after sorting ith row of adjacency matrix. I sort every row of adjacency matrix, after doing that program’s time complexity reduced from O(n3) to O(n2).

After creating sorted adjacency matrix, program starts to find tours with nearest neighbor algorithm, putting each city as first city on tour one by one. To find nearest neighbor, program just looks first few index of adjacency matrix’s corresponding rows, if any city in jth index is added before, program looks for j+1st  index until finding city that not added before and returns that city and distance between given ith and found city.

The city returns from ***find\_nearest(i)*** method will be added to current tour, after completing current tour, ***nearest\_neighbor(first\_city)***returns it and its total length.

Program compares returned tour’s length to current best tour’s length and if it’s length shorter than before, returned tour will be assigned to current best tour.

Lastly, my program writes found best tour and its length to output file named out.txt and finishes.

My program runs in O(n2) time complexity as I mentioned before;

python hw3.py test-input-1.txt

2881

0.20804715156555176

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

python hw3.py test-input-2.txt

304268

3.31174373626709

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

python hw3.py test-input-4.txt

12017

29.711667776107788

↑ This was reduced from ~2750s by implementing presorting on adjacency matrix.

|  |  |  |  |
| --- | --- | --- | --- |
| **Total City Number** | **Execution Time** | **Num x:**  **(i+1)/(i)** | **Time y:**  **(i+1)/(i)** |
| 280 | 0.208 | - | - |
| 1002 | 3.312 | 3.58 | 15.92 |
| 2924 | 29.712 | 2.92 | 8.97 |

3.58 x 3.58 = 12.82 ~15.92

2.92 x 2.92 = 8.53 ~~8.97

\* Program gives memory error while trying to get an output from largest test input. So I am not able to add output file of test-input-3.txt because of number of cities.

\*\* Doğrulama programında benim programımın çıktılarını denerken ‘city not found’ hatası alıyorum, çünkü çözüm turunda ilk ve son şehirler aynı olduğundan ve doğrulama programı çıktıdaki şehirleri küçükten büyüğe sıraladıktan sonra bir döngüde indeks sayısı ile o indekse karşılık gelen şehrin kimlik numarasını karşılaştırırken ilk şehirden 2 adet bulunmasından dolayı ilk şehrin indeksinden sonrasında listenin sonuna kadar hata mesajı yazdırıyor. Bu durumu paylaşmak istedim.