Seles Traveling Problem

➢ P=NP ? Perhaps, why not?

Let's start with the traveling salesman problem. Suppose he is visiting 15 countries, and therefore, we need to try all possibilities to find the shortest path that passes through all 15 countries. But why should we try all possibilities? Why not try only one possibility?

With continuous experimentation, I noticed that the shortest path was always the path with the closest sum of the first three points to it. Thus, let's start by determining the starting point. Let's assume we have a 15x15 matrix containing the distances between cities. We will sort the matrix and then compare the first three elements of each row(without 0) and extract the row with the smallest sum. In case of similarity, we will take the element following it until we obtain the smallest row. The outer row is the starting point, let's assume it is 7.

But what's next? Secondly, we want to determine the second point. The second point will have a number of possibilities equal to the number of elements that were previously compared (usually three unless there was a similarity). Let's assume there is no similarity, then we have three paths that share the starting point (7) and differ in the next point. The next point will be determined based on the column where the smallest value is located (we now have three smallest values in three paths). The first path will start at 7 and the next point will be determined by finding the column where the smallest element in row 7 is located (let's assume it is in the third column). Then, the next point will be 3, and we repeat the same process in the third row to find the next row.

If the smallest value in the row leads to a row that has been visited before, we will not take it and take the next smallest value. We repeat this process for the other two paths that have already determined their starting and second points.

We will see the shortest path taken in the three paths, and this will be the shortest path that can be taken without trying all the unproductive possibilities.

If we find two identical elements in searching for the smallest element in a row, then the path will have two overlapping paths in previous points but different in the next steps.

Let's take an example. Let's assume we have distances between each of the 15 countries. We will have a 15x15 matrix containing these distances.

```
Distance=([[ 0, 48, 65, 68, 68, 10, 84, 22, 37, 88, 71, 89, 89, 13, 59],
           [66,  0, 88, 47, 89, 82, 38, 26, 78, 73, 10, 21, 81, 70, 80],
           [48, 65,  0, 89, 50, 30, 20, 20, 15, 40, 33, 66, 10, 58, 33],
           [32, 75, 24,  0, 76, 56, 29, 35,  1,  1, 37, 54,  6, 39, 18],
           [80,  5, 43, 59,  0,  2, 66, 42, 58, 36, 12, 47, 83, 92,  1],
           [15, 54, 13, 43, 85,  0, 69,  7, 69, 48,  4, 77, 53, 79, 16],
           [21, 59, 24, 80, 14, 86,  0, 50, 70, 42, 36, 65, 96, 70, 95],
           [ 1, 51, 37, 35, 49, 94,  4,  0, 43, 78, 22, 74,  1, 11, 44],
           [59, 24, 60,  3, 99, 63, 36, 95,  0, 83, 47, 21, 82, 51, 28],
           [15, 42, 59, 66, 37, 11, 87, 44, 12,  0, 52, 81, 33, 55,  1],
           [39, 20, 47, 43, 57, 61, 78, 31, 25,  3,  0, 95, 99, 14, 41],
           [73, 20, 96, 73, 27, 67, 53, 68, 62, 15, 97,  0, 68, 12, 87],
           [78, 76, 57, 17, 25, 30, 22, 26, 81, 61, 62, 84,  0, 33, 71],
           [86, 32, 14, 72, 57, 25, 80, 42, 19, 41, 55, 80, 12,  0, 94],
           [ 2, 96, 45, 89, 25, 68, 83,  4, 77, 36, 87, 62, 70, 88,  0]])
```

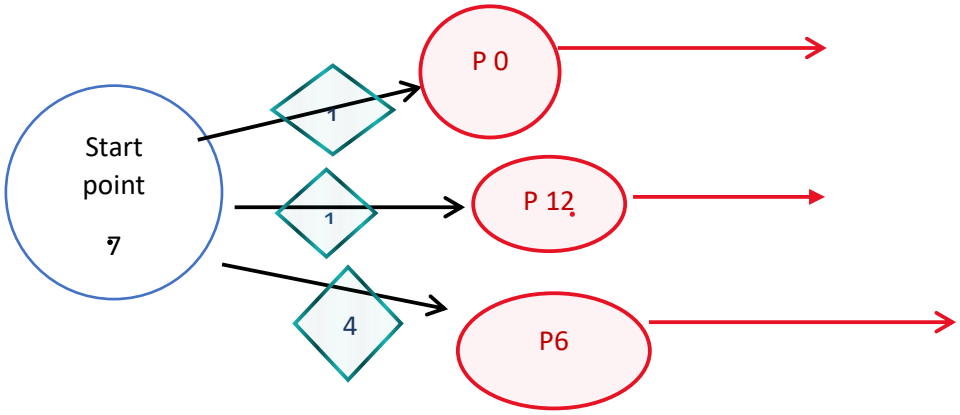
Secondly, let's sort it.

```
distances=[[ 0, 10, 13, 22, 37, 48, 59, 65, 68, 68, 71, 84, 88, 89, 89],
           [ 0, 10, 21, 26, 38, 47, 66, 70, 73, 78, 80, 81, 82, 88, 89],
           [ 0, 10, 15, 20, 20, 30, 33, 33, 40, 48, 50, 58, 65, 66, 89],
           [ 0,  1,  1,  6, 18, 24, 29, 32, 35, 37, 39, 54, 56, 75, 76],
           [ 0,  1,  2,  5, 12, 36, 42, 43, 47, 58, 59, 66, 80, 83, 92],
           [ 0,  4,  7, 13, 15, 16, 43, 48, 53, 54, 69, 69, 77, 79, 85],
           [ 0, 14, 21, 24, 36, 42, 50, 59, 65, 70, 70, 80, 86, 95, 96],
           [ 0,  1,  1,  4, 11, 22, 35, 37, 43, 44, 49, 51, 74, 78, 94],
           [ 0,  3, 21, 24, 28, 36, 47, 51, 59, 60, 63, 82, 83, 95, 99],
           [ 0,  1, 11, 12, 15, 33, 37, 42, 44, 52, 55, 59, 66, 81, 87],
           [ 0,  3, 14, 20, 25, 31, 39, 41, 43, 47, 57, 61, 78, 95, 99],
           [ 0, 12, 15, 20, 27, 53, 62, 67, 68, 68, 73, 73, 87, 96, 97],
           [ 0, 17, 22, 25, 26, 30, 33, 57, 61, 62, 71, 76, 78, 81, 84],
           [ 0, 12, 14, 19, 25, 32, 41, 42, 55, 57, 72, 80, 80, 86, 94],
           [ 0,  2,  4, 25, 36, 45, 62, 68, 70, 77, 83, 87, 88, 89, 96]]
```

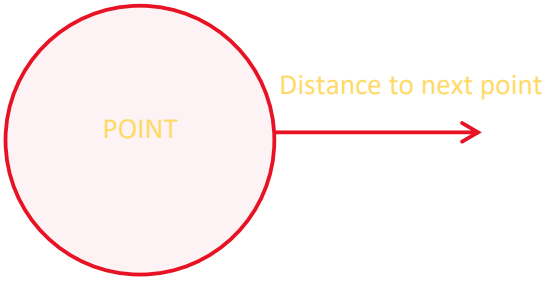
Then, we add the first three elements in each row without including the first element (0), and compare to obtain the smallest row (starting point).

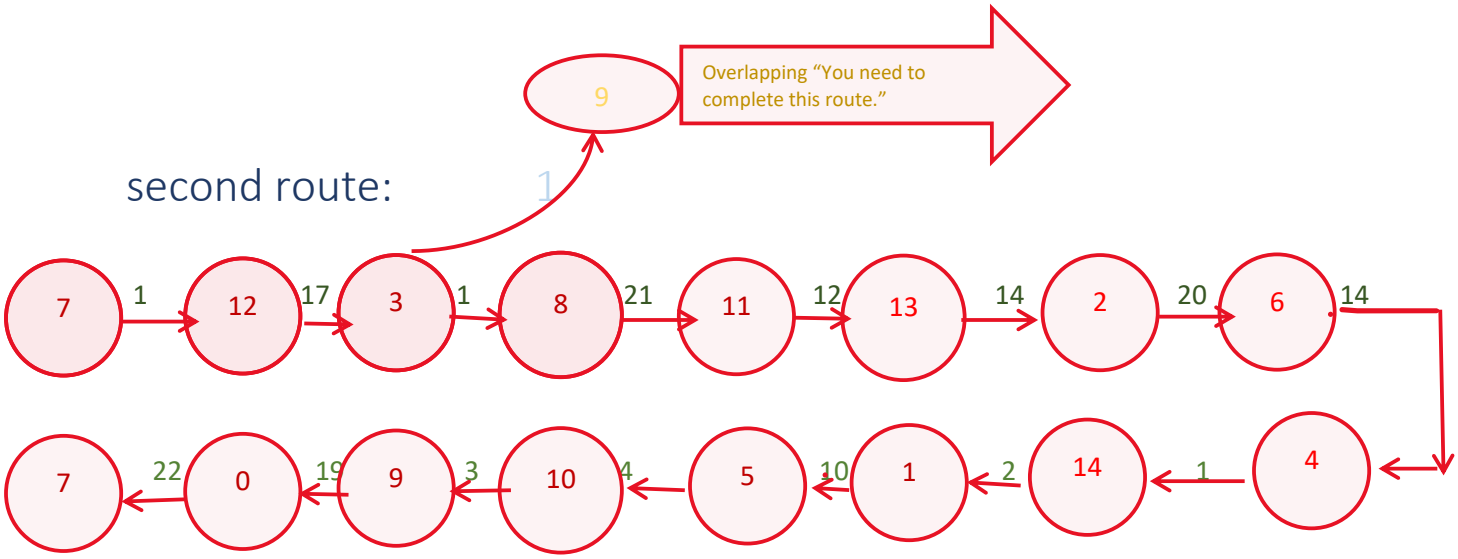
```
smallest_row = 0
for i in range(len(distances)):
    if distances1[i][:3] < distances1[smallest_row][:3]:
        smallest_row = i
    elif distances1[i][:3] == distances1[smallest_row][:3]:
        if distances1[i][3] < distances1[smallest_row][3]:
            smallest_row = i

print("The smallest row is:", smallest_row)



The smallest row is: 7
```

It is possible to extract the last element used in the comparison because we need to determine the paths for all points included in the comparison (in cases where some rows have the same value, as we mentioned earlier).



As we mentioned earlier, we go back to the original matrix and search for the number 1, then extract the number of the column where it is located, which will be the number of the next row, and so on for all paths.

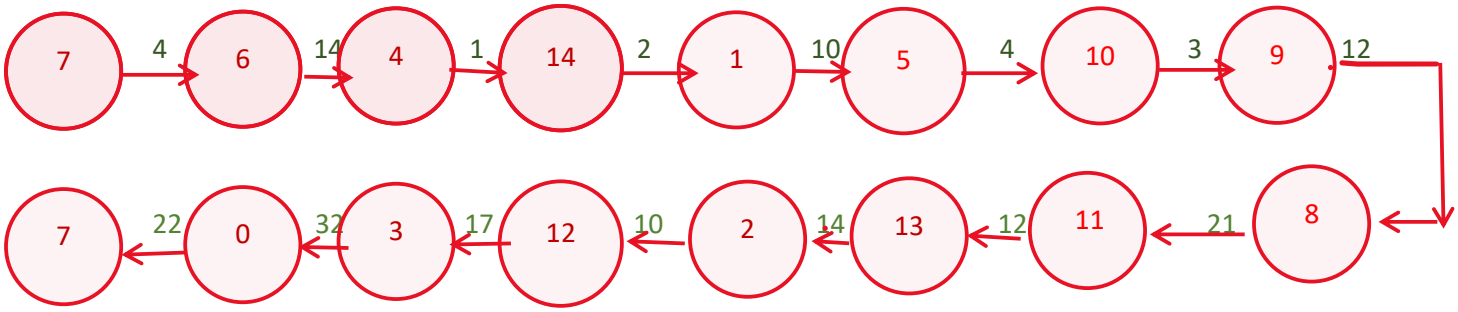




Total distance=1+17+1+21+12+14+20+14+1+2+10+4+3+19+22=161

Cities visited in order[7,12,3,8,11,13,2,6,4,14,1,5,10,9,0,7]

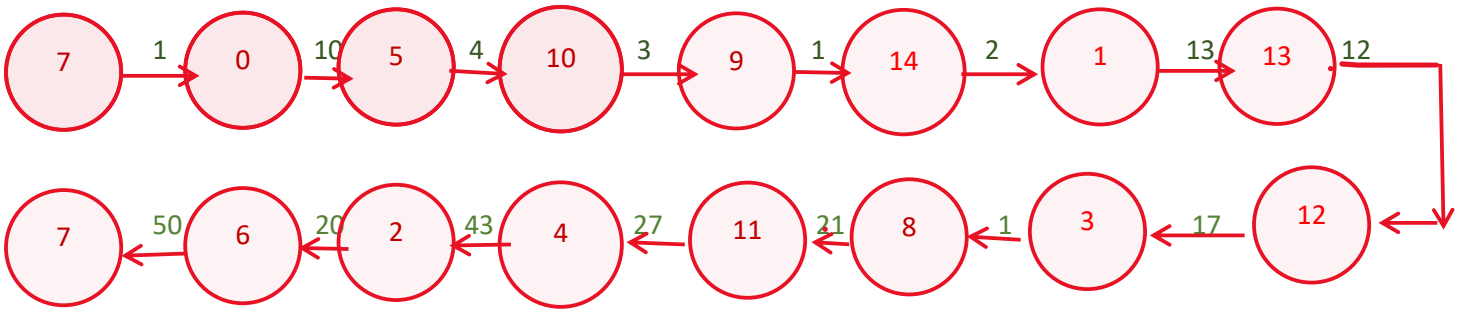
Third route:



Total distance=4+14+1+2+10+4+3+12+21+12+14+10+17+32+22=178

Cities visited in order[7,6,4,14,1,5,10,9,8,11,13,2,12,3,0,7]

First route:



Total distance=1+10+4+3+1+2+13+12+17+1+21+27+43+20+50=225

Cities visited in order[7,0,5,10,9,144,1,13,12,3,8,11,4,2,6,7]

We notice that the second route has the shortest distance, which is what is required, without trying the remaining solutions, because this is the correct solution. Also, don't forget to complete the sub-route that appeared when the same minimum value was repeated. It should be calculated, so we will have four routes instead of three, and we will compare between them.

Good luck

ENG/Mohamed zonkol

Mo.zonkol@gmail.com