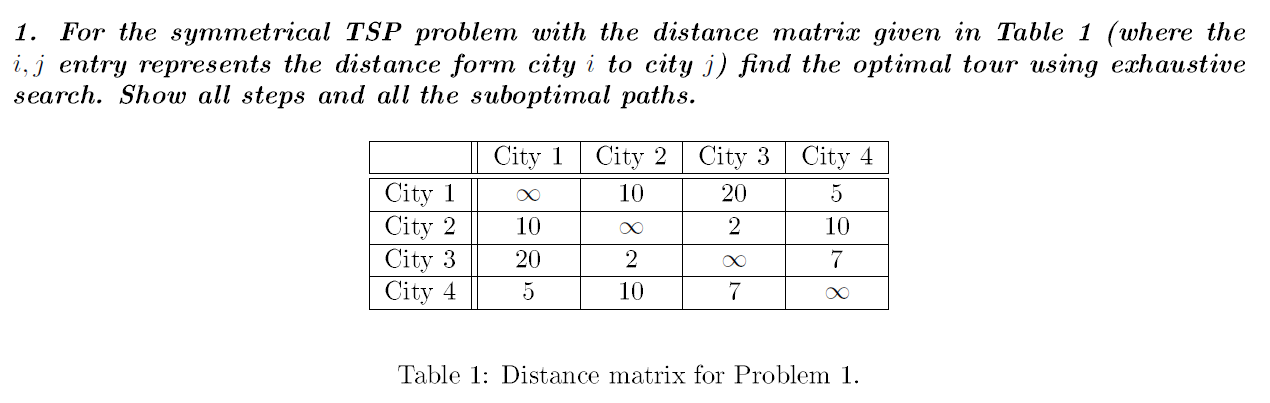
**CSE620: Assignment2**

**Sima Shafaei**

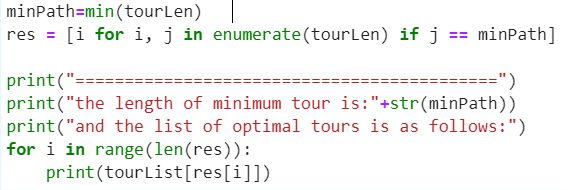
**Due: November 6, 2020**



***Solution:***

*We wrote a Python program for this question:*





*The result is as follows:*

tour: city1-->city2-->city3-->city4-->city1 length:24

tour: city1-->city2-->city4-->city3-->city1 length:47

tour: city1-->city3-->city2-->city4-->city1 length:37

tour: city1-->city3-->city4-->city2-->city1 length:47

tour: city1-->city4-->city2-->city3-->city1 length:37

tour: city1-->city4-->city3-->city2-->city1 length:24

tour: city2-->city1-->city3-->city4-->city2 length:47

tour: city2-->city1-->city4-->city3-->city2 length:24

tour: city2-->city3-->city1-->city4-->city2 length:37

tour: city2-->city3-->city4-->city1-->city2 length:24

tour: city2-->city4-->city1-->city3-->city2 length:37

tour: city2-->city4-->city3-->city1-->city2 length:47

tour: city3-->city1-->city2-->city4-->city3 length:47

tour: city3-->city1-->city4-->city2-->city3 length:37

tour: city3-->city2-->city1-->city4-->city3 length:24

tour: city3-->city2-->city4-->city1-->city3 length:37

tour: city3-->city4-->city1-->city2-->city3 length:24

tour: city3-->city4-->city2-->city1-->city3 length:47

tour: city4-->city1-->city2-->city3-->city4 length:24

tour: city4-->city1-->city3-->city2-->city4 length:37

tour: city4-->city2-->city1-->city3-->city4 length:47

tour: city4-->city2-->city3-->city1-->city4 length:37

tour: city4-->city3-->city1-->city2-->city4 length:47

tour: city4-->city3-->city2-->city1-->city4 length:24

*Based on above result the length of minimum tour is* ***24*** *and the list of all optimal tours is:*

city1-->city2-->city3-->city4-->city1

city1-->city4-->city3-->city2-->city1

city2-->city1-->city4-->city3-->city2

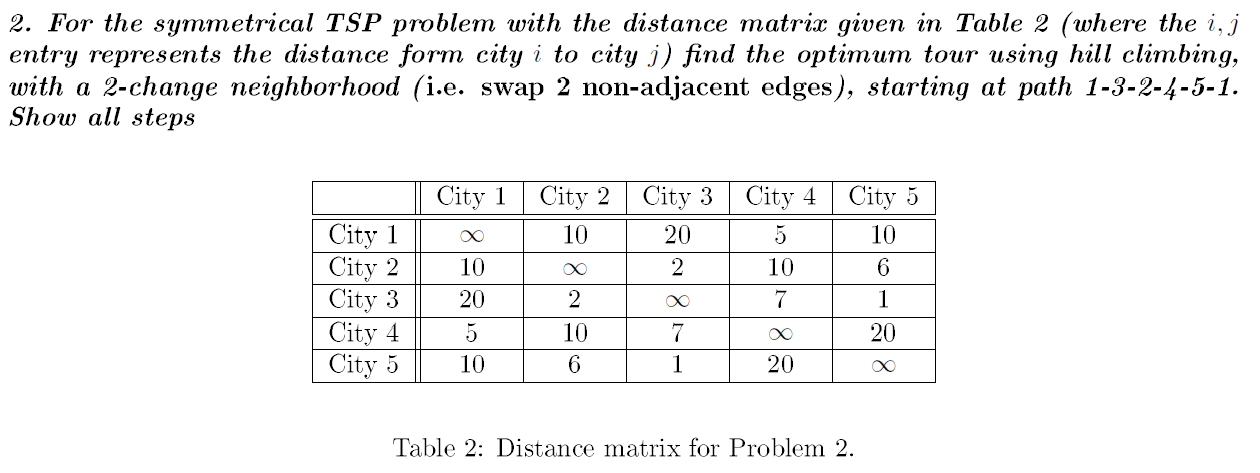
city2-->city3-->city4-->city1-->city2

city3-->city2-->city1-->city4-->city3

city3-->city4-->city1-->city2-->city3

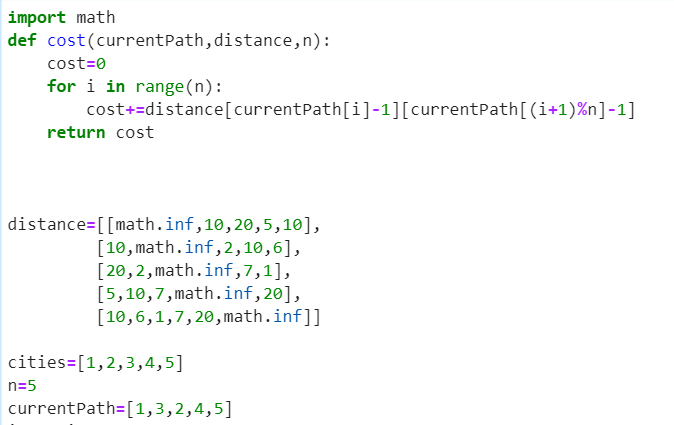
city4-->city1-->city2-->city3-->city4

city4-->city3-->city2-->city1-->city4



***Solution:***

*To solve this problem again we create a python program:*





*The initial path is :[1,3,2,4,5,1], with Cost = 62.*

*The first group of neighbors in first iteration, and their cost are:*

swap:1->3 and 2->4: path= [1, 2, 3, 4, 5,1] cost=49

swap:1->3 and 4->5: path= [1, 4, 2, 3, 5,1] cost=28

swap:3->2 and 5->1: path= [1, 3, 5, 4, 2,1] cost=48

swap:2->4 and 1->3: path= [4, 3, 2, 1, 5,4] cost=36

swap:4->5 and 3->2: path= [1, 5, 2, 4, 3,1] cost=53

*The path*=[1, 4, 2, 3, 5,1] *with cost=28 has the minimum weight and will be selected for the next iteration. The neighbors of selected path in second iteration are:*

swap:1->4 and 2->3 [1, 2, 4, 3, 5] cost=38

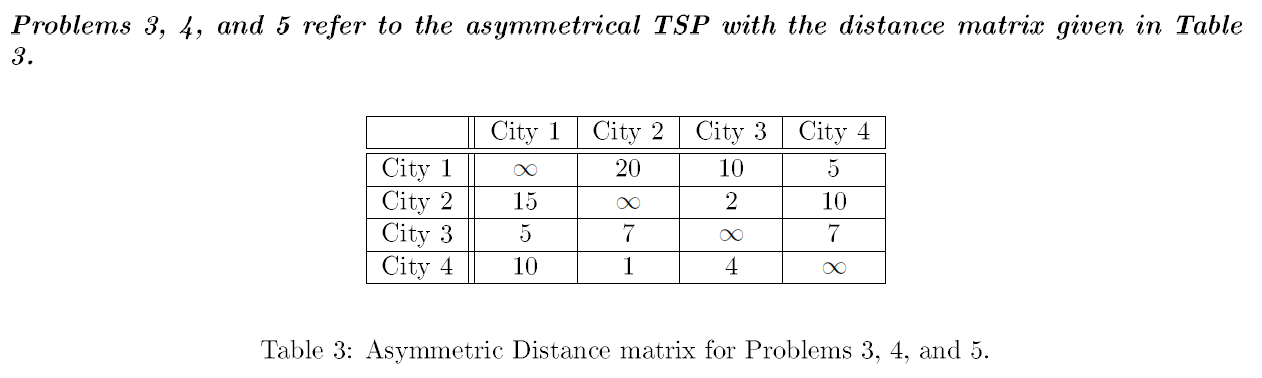
swap:1->4 and 3->5 [1, 3, 2, 4, 5] cost=62

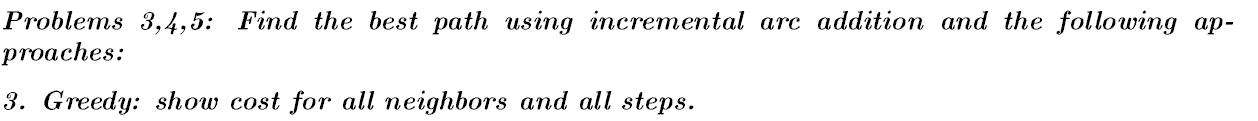
swap:4->2 and 5->1 [1, 4, 5, 3, 2] cost=38

swap:2->3 and 1->4 [3, 4, 2, 1, 5] cost=38

swap:3->5 and 4->2 [1, 5, 2, 3, 4] cost=30

*As we can see there is no path with cost less than 28, so we can stop the search and the optimum solution would be : path*=[1, 4, 2, 3, 5,1] *and Cost=28*





***Solution:***

*in first iteration we have to select the first edge and select the one with minimum weight:*



*In next steps, in each iteration, we have to add one node to our connected path that would not create a loop and select the minimum weight among all possible paths*.

*In this iteration we have two winners (1-4 and 3-4). We selected the first one in the list.*

*Among all possible selection 4->2 has the minimum weight and will be selected in the first iteration.*

*.Iteration 2:*



*Iteration 3:*



*In this iteration the winner is 2->3*

*In this iteration winner is 2->3. Finally, when we added all the nodes we have to return to the origin therefore we will add 3->1 to complete the tour:*



*The total length of the final tour is 13.*



***Solution:***

* *Estimating Lower Bound*

Subtract each row from min

Find min for each row

|  |
| --- |
| *Min* |
| 5 |
| 2 |
| 5 |
| 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | 15 | 5 | 0 |
| *City 2* | 13 | ∞ | 0 | 8 |
| *City 3* | 0 | 2 | ∞ | 2 |
| *City 4* | 9 | 0 | 3 | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | 20 | 10 | 5 |
| *City 2* | 15 | ∞ | 2 | 10 |
| *City 3* | 5 | 7 | ∞ | 7 |
| *City 4* | 10 | 1 | 4 | ∞ |

Find min for each column

LB=

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | 15 | 5 | 0 |
| *City 2* | 13 | ∞ | 0 | 8 |
| *City 3* | 0 | 2 | ∞ | 2 |
| *City 4* | 9 | 0 | 3 | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Min | 0 | 0 | 0 | 0 |

LB=

Subtract each con from its min

=================================================================================

*Most influential arc:*

1->4 : LB+(5+2)

2->3: LB+(8+3) Max

3->1: LB+(2+9) Max

4->2: LB+(3+2)

LB=13+11=24

We can choose either 3->1 or 2->3 for branching. We will select 3->1

================================================================================

***include 3->*1**

Find min for each row

|  |
| --- |
| *Min* |
| 0 |
| 0 |
|  |
| 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | 15 | ∞ | 0 |
| *City 2* | ∞ | ∞ | 0 | 8 |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | 0 | 3 | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | 15 | ∞ | 0 |
| *City 2* | ∞ | ∞ | 0 | 8 |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | 0 | 3 | ∞ |

Find min for each column

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | 15 | ∞ | 0 |
| *City 2* | ∞ | ∞ | 0 | 8 |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | 0 | 3 | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Min | ∞ | 0 | 0 | 0 |

Subtract each con from its min

LB=13+0=13



=================================================================================

*Most influential arc:*

1->4 : LB+(15+8) Max

2->3: LB+(8+3)

4->2: LB+(3+15)

1->4 will be selected

LB=13+23=36

================================================================================

***include 1->*4**

Find min for each row

|  |
| --- |
| *Min* |
|  |
| 0 |
|  |
| 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | ∞ | ∞ | ∞ |
| *City 2* | ∞ | ∞ | 0 | ∞ |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | 0 | 3 | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | ∞ | ∞ | ∞ |
| *City 2* | ∞ | ∞ | 0 | ∞ |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | 0 | 3 | ∞ |

Find min for each column

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | ∞ | ∞ | ∞ |
| *City 2* | ∞ | ∞ | 0 | ∞ |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | 0 | 3 | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Min | ∞ | 0 | 0 | 0 |

Subtract each con from its min

LB=13+0=13



=================================================================================

*Most influential arc:*

We can choose either 4->2 or 2->3 for branching. We will select 4->2

LB=13+3=16

2->3: LB+(3)

4->2: LB+(3)

================================================================================

***include 4->*2**

Find min for each row

|  |
| --- |
| *Min* |
|  |
| 0 |
|  |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | ∞ | ∞ | ∞ |
| *City 2* | ∞ | ∞ | 0 | ∞ |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | ∞ | ∞ | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | ∞ | ∞ | ∞ |
| *City 2* | ∞ | ∞ | 0 | ∞ |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | ∞ | ∞ | ∞ |

Find min for each column

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *City 1* | *City 2* | *City 3* | *City 4* |
| *City 1* | ∞ | ∞ | ∞ | ∞ |
| *City 2* | ∞ | ∞ | 0 | ∞ |
| *City 3* | ∞ | ∞ | ∞ | ∞ |
| *City 4* | ∞ | ∞ | ∞ | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Min |  |  | 0 |  |

Subtract each con from its min

LB=13+0=13



================================================================================

*We need only one more arc to back to the first city and it would be 2->3. Therefore, the optimal path is*: **3->1->4->2->3**



***Solution:***

*The general formula that followed by dynamic programming for TSP is:*

*In this problem we start with city 1 therefore, i=1 and S={2,3,4} and*

*We start with creating tree and compute it from bottom to top, following tree shows all possible tour in four cities:*



***First level in tree:***



***Second level in the tree:***



***third level in the tree:***

*(minimum come from k=3 or 4)*

*(minimum come from k=4)*

*(minimum come from k=2)*



***The root of tree:***

*(minimum come from k=4)*



Based on where minimum came from in each step the shortest tour is: 1->4->2->3->1