Consider an undirected complete graph with *n* = 5 nodes. The length of the links can be represented by a symmetric matrix as follows:

1 2 3 4 5

1 X 3 4 2 7

2 3 X 8 1 11

3 4 8 X 6 4

4 2 1 6 X 3

5 7 11 4 3 X

In this example, the length of link {2, 4} is 1. The ‘X’ would represent infinity. The *Traveling Salesman Problem* (TSP) is to find the shortest possible *tour* in the graph. That is, a single simple cycle that starts at a node, visits every other node exactly once, and returns to the starting node.

The tour (1, 2, 3, 4, 5, 1), which uses links {1,2}, {2,3}, {3,4}, {4,5} and {5,1}, would have a total length of 3 + 8 + 6 + 3 + 7 = 27.

The tour (1, 4, 2, 5, 3, 1) has a total length of 2 + 1 + 11 + 4 + 4 = 22.

A “brute force” method to solve the problem is to calculate the total length of each of the (*n*-1)!/2 possible tours and identify the shortest. This would be terribly inefficient, especially for more realistic networks which might have hundreds or even thousands of nodes. Unfortunately, there is currently no known method that finds the optimal solution to the TSP in a number of steps that is bounded by a polynomial in *n*.

We instead seek a *heuristic method* that finds a relatively short tour in a reasonable number of steps. For example, the *Nearest Neighbor Method* is a “greedy” scheme for a complete graph that works as follows:

1. Start at node 1
2. For steps := 1 to n-1

Travel on the shortest link from the current node to an **unvisited** node

1. Travel on the link from the last node back to node 1

**Part I**

1. What is the complexity of the Nearest Neighbor Algorithm on a complete graph?

1. Perform the method to obtain a solution for the complete graph above and give its total cost.
2. Perform the method, starting at a different node instead of 1, to obtain a solution and give its total cost. It is possible to obtain a tour that is different from the one you found in part a). Give a reason why this can happen.

**Part II**

There are a variety of additional heuristic approaches that can be developed to find a reasonable solution to the TSP. In this part, you will implement an “iterative improvement” approach, which starts with some solution, like one obtained from the nearest neighbor heuristic, and attempts to find a cheaper solution by making a simple change.

In a “Two-Swap” approach, two links in the solution, say {a,b} and {c,d}, are removed and replaced by {a,c} and {b,d}, (or by {a,d} and {b,c}). If this new solution has a shorter total length, then the changes are kept and we continue to try to improve. Otherwise we place {a,b} and {c,d} back and try to improve with a different pair of links to replace. We continue this process until we believe no such adjustment will yield any improvements, (e.g., when all link pairs are tried without any improvement).

You will then apply this method to the following problem instance. You can start with any solution you like

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **From \ To** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **1** | X | 13 | 12 | 9 | 11 | 7 | 2 | 7 |
| **2** | 13 | X | 10 | 10 | 13 | 5 | 12 | 7 |
| **3** | 12 | 10 | X | 5 | 24 | 17 | 13 | 10 |
| **4** | 9 | 10 | 5 | X | 12 | 19 | 15 | 21 |
| **5** | 11 | 13 | 24 | 12 | X | 3 | 4 | 9 |
| **6** | 7 | 5 | 17 | 19 | 3 | X | 14 | 17 |
| **7** | 2 | 12 | 13 | 15 | 4 | 14 | X | 3 |
| **8** | 7 | 7 | 10 | 21 | 9 | 17 | 3 | X |

**Hand in:**

1. Give your starting solution and its total length.
2. Attempt at least 5 “swaps”. Give your final solution and its total length.

**Extra Credit:** Explain why it is possible that two different students performing this algorithm are likely to obtain two different final solutions.