Homework Assignment 5

Your assignment for Homework 5 is to write a simple recursive program to solve the Travelling Salesman Problem (TSP).

The TSP and its variants are among of the classic problems in computing. Because TSP is an NP-complete problem, we don't have a good solution for it, and we have to live with heuristics and approximations.

Finding a polynomial-time solution to the TSP would win you a million dollar prize from the Clay Institute. It's that big a problem.

You have been given a matrix of cities and pairwise distances between cities. Assume you are a salescreature based in Seattle and this is the list of cities you must visit in a cyclic tour of all the cities, with a return home to Seattle. Because distance costs money, the goal in the TSP is to find a cyclic path through the set of cities that requires the minimum total distance. (In the TSP, "distance" and "cost" are synonyms.)

You are to write a program that recursively generates all the permutations that constitute cycles through the cities, computes the distance of a complete cycle, and eventually outputs the cycle that has the minimal total distance.

The problem assumes that you are starting and ending your tour in Seattle, so there are $6! = 720$ possible cycles to look at. I recommend you trim this down for testing purposes by deleting two cities from your matrix. The resulting matrix will have $4! = 24$ possible cycles, and that's small enough for you to examine by eye.

NOTE: It is possible to determine that a partial permutation cannot be completed to an optimal complete permutation because the partial permutation is already more costly than the current best permutation. If that's the case, then the computation can be made more efficient by an early abort strategy. You do not need to do this. That is, you can wait to make the decision on whether the full permutation is a new best solution until the complete permutation is generated.