

## Problem

Given a number of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

## Algorithm

Starting at first city, mark it as visited, next go to closest city, and repeat what you did in starting city do this until all connecting cities are marked as visited.

## Program Design

This design first creates an adjacency matrix with a given a text file. Second creates a stack to record the solution, and an array to record visited cities. Third it processes the starting city by adding it to the visited cities and to the path it also prints the city. Fourth in a while loop the cities are selected by shortest path. Fifth the third and fourth steps are repeated until all connecting cities have been processed.

## Results

run:

Best path for 12

0 5 3 8 4 1 11 6 7 10 9 2

BUILD SUCCESSFUL (total time: 0 seconds)

run:

Best path for 14

0 5 3 8 4 1 13 11 6 7 10 9 2 12

BUILD SUCCESSFUL (total time: 0 seconds)

run:

Best path for 19

0 5 11 8 4 1 9 3 14 18 15 12 7 6 10 13 17 16 2

BUILD SUCCESSFUL (total time: 0 seconds)

Best path for 29

0 27 5 11 8 4 20 1 19 9 3 14 18 24 6 22 26 23 7 15 12 17 13 21 16 10 28 25 2

BUILD SUCCESSFUL (total time: 0 seconds)

## Observations

This Algorithm does not have the problem of having an exponential, compared to  $N$ , run time that depth first search algorithm suffers from, but it is unlikely to give the best answer to the Traveling Salesperson Problem.