Traveling Salesman

This is an example of BestFirst(Fit) Search algorithm
Greedy Algorithm
Branch and Bound

Hamiltonian circuit
Every vertex uses exactly once.

Brute force you get to select from n! paths.

Representation M(i,j) as distance from i to j.
M(i,i) is infinite, no self-edges, in reality we use M(i,i)=0 for min distance, but for convenience we use infinity to indicate not self-edge.

If any row (or column) is reduced , the total length of circuit is reduced by that amount, because exactly one edge on a path is reduced.

Subtracting the rows and columns give us an estimate of **lower bond** on the length of traveling salesman path. Reduce such that each row has at least one zero to further simplify the problem

If an edge is included, not other edge with same start or end vertex can be on the path, that is, if (i,j) is an edge in the path, the (i,*) and (*,j) cannot be on the path. This is the reason for deleting the row and column in the matrix. This reduces the matrix to a smaller size for further consideration.
If (i,j) is not an edge on a path, its length is set to infinity from further consideration.

Experiments on nxn matrices indicate that complexity is $O(1.26^n)$

Procedure
For each row decrease by min(>o) of the row
For each column decrease by min(>0) of the column
Add these amounts of decrease to get lower bound on the estimate of shortest path.

Now each row has at least one zero and each column has at least one zero
Start with an edge corresponding any zero
Create two matrices, one that includes this edge and one that excludes this edges
Apply the reduction to these two matrices and update the lower bounds and follow the smaller lower bound. Note if (v_i,v_j) was the last edge considered, now try the $(v_j,*)$ edge for next consideration.
Keep updating the lower bound,
Keep branches that include and that exclude a particular edge. Follow the branch with overall smaller lower bound.
Keep doing this until you end up with a full cycle.