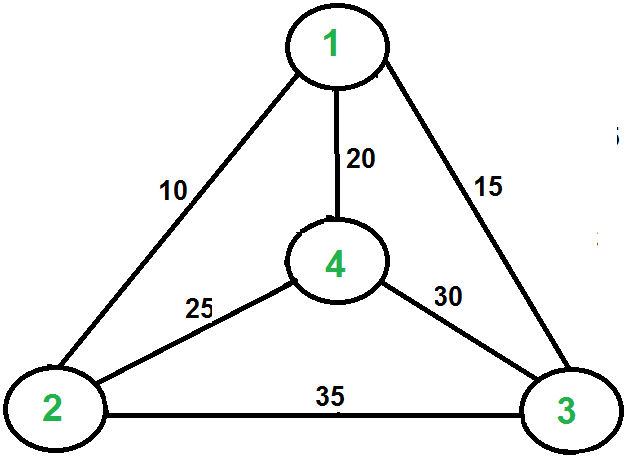
# Problem Statement:

**Travelling Salesman Problem (TSP):** Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.

Note: Here we need to find a minimum weight Hamiltonian Cycle.  


For example:  consider the graph shown in the above figure. A TSP tour in the graph is 1-3-4-2-1. The cost of the tour is 15+30+25+10= 80.

# Solution:

## Algorithm(Using Back Tracking):

* Consider city 1 (let say 0th node) as the starting and ending point. Since route is cyclic, we can consider any point as starting point.
* Start traversing from the source to its adjacent nodes in dfs manner.
* Calculate cost of every traversal and keep track of minimum cost and keep on updating the value of minimum cost stored value.
* Return the permutation with minimum cost.