Traveling Salesperson Problem

Criteria:

- Salesperson travels between N cities
- Salesperson finishes where he/she started
- Salesperson visits each city exactly once (but the starting/finishing city)
- All cities are connected to each other directly
- Each node (way between cities) has a weight: its difficulty to travel using this node

Goal: Find the easiest/shortest route

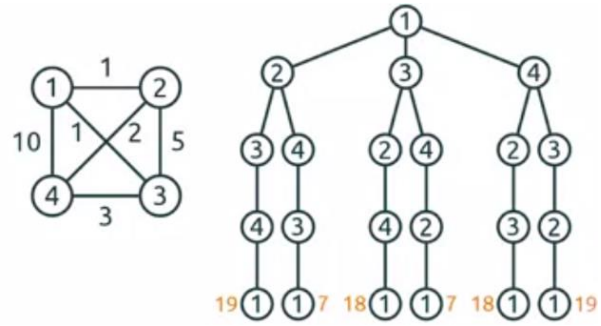
Algorithms:

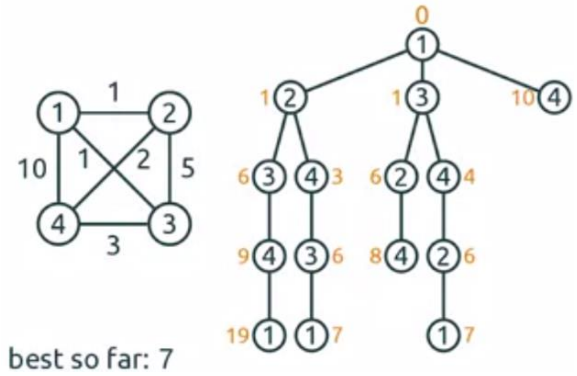
The easiest but most expensive solution is to simply try all possibilities. (for N cities you get (N-1)! possibilities). This means that for only 10 cities there are over 180 thousand combinations to try (since you can define the starting city, there can be variations on the remaining nine). We only count half since each route has an equal route in reverse with the same length or cost.

- (10-1)! /2 = 181440

Branch and bound:

The algorithm consists of a systematic listing of possible solutions. It is thought to form a rooted tree with the full set at the root. The algorithm checks the branches of the tree, which represent the subsets of the solution sets.





It then saves the best solution calculated so far and compares it against upcoming solutions to find the optimal answer.

If there are n nodes there will be (n-1)! feasible solutions.

Branch and bound can be optimized with heuristic[1] approaches. We used the approach suggested by Prof. Abdul Bari. You can visit https://www.youtube.com/watch?v=1FEP_sNb62k for further information where he explains the algorithm step by step.

[1] Heuristics approaches use a set of guiding rules to select the next node to visit. But heuristics result in approximations, so there is no guarantee to get the optimal solution. However, they can find useful solutions in a fraction of the time required by a brute force approach.