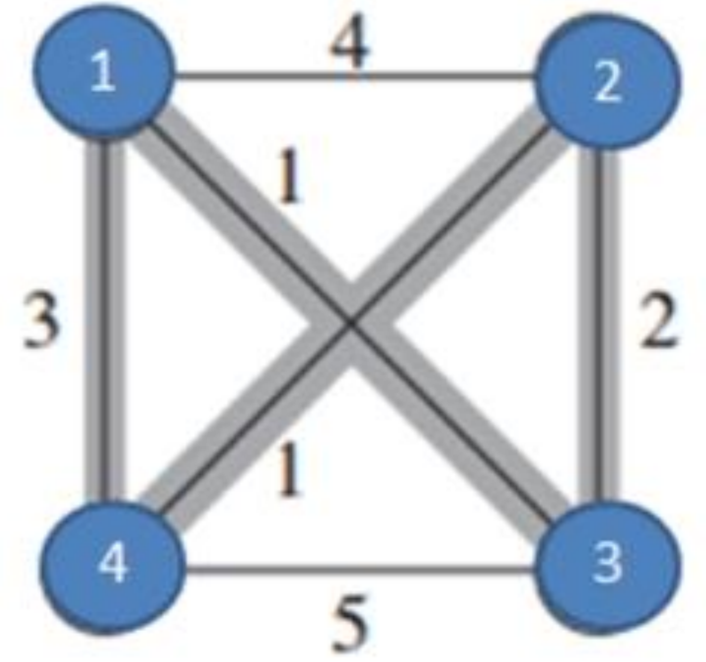
Unit 4: Dynamic Programming

Travelling Salesman Problem



$$
\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 4 & 1 & 3 \\
2 & 4 & 0 & 2 & 1 \\
3 & 1 & 2 & 0 & 5 \\
4 & 3 & 1 & 5 & 0
\end{array}
$$

Lets start from node 1

- T (1, {2,3,4}) = minimum of
- { (1,2) + T (2, {3,4}) 4+**6**=10
- = { (1,3) + T (3, {2,4}) 1+**3**=4
- = { (1,4) + T (4, {2,3}) 3+**3**=6

- Here minimum of above 3 paths is answer but we know only values of (1,2) , (1,3) , (1,4) remaining thing which is T (2, {3,4}) …are new problems now.
- First we have to solve those and substitute here.
  - T (2, {3,4}) = minimum of
    - = { (2,3) + T (3, {4}) 2+**5**=7
    - = { (2,4) + T {4, {3}) 1+**5**=6

- T (3, {2,4}) = minimum of
  - = { (3,2) + T (2, {4}) 2+**1**=3
  - = { (3,4) + T {4, {2}) 5+**1**=6
- T (4, {2,3}) = minimum of
  - = { (4,2) + T (2, {3}) 1+**2**=3
  - = { (4,3) + T {3, {2}) 5+**2**=7
- T (3, {4}) = (3,4) + T (4, {}) 5+0=5
- T (4, {3}) = (4,3) + T (3, {}) 5+0=5
- T (2, {4}) = (2,4) + T (4, {}) 1+0=1
- T (4, {2}) = (4,2) + T (2, {}) 1+0 = 1
- T (2, {3}) = (2,3) + T (3, {}) 2+0 = 2
- T (3, {2}) = (3,2) + T (2, {}) 2+0=2
- Here T (4, {}) is reaching base condition in recursion, which returns 0 (zero) distance.
- This is where we can find final answer,

- T (1, {2,3,4}) = minimum of
  - = { (1,2) + T (2, {3,4}) 4+**6**=10 in this path we have to add +1 because this path ends with 3. From there we have to reach 1 so 3->1 distance 1 will be added total distance is 10+1=11
  - = { (1,3) + T (3, {2,4}) 1+**3**=4 in this path we have to add +3 because this path ends with 3. From there we have to reach 1 so 4->1 distance 3 will be added total distance is 4+3=7
  - = { (1,4) + T (4, {2,3}) 3+**3**=6 in this path we have to add +1 because this path ends with 3. From there we have to reach 1 so 3->1 distance 1 will be added total distance is 6+1=7
  - Minimum distance is **7** which includes path **1->3->2->4->1.**

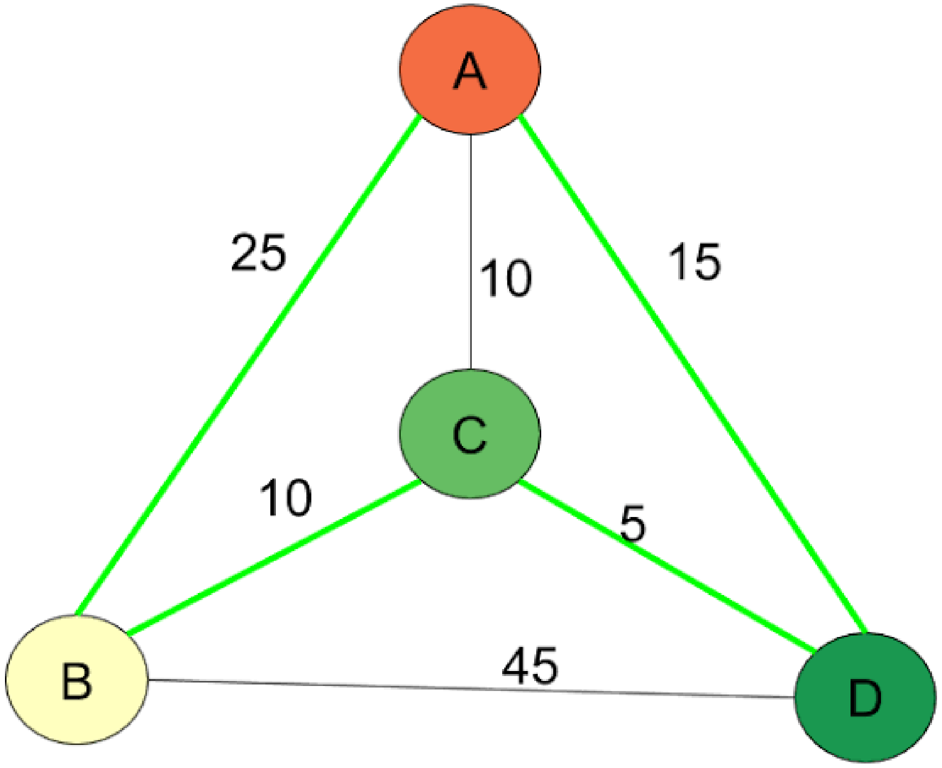
Formula

- **T (i , s) = min ((i , j) + T (j , S – { j })) ;** S!= Ø ; j € S ;
  - S is set that contains non visited vertices

    = (i, 1) ; S=Ø, This is base condition for this recursive equation.

Here,

  - T (i, S) means We are travelling from a vertex “i” and have to visit set of non-visited vertices “S” and have to go back to vertex 1 (let we started from vertex 1).
  - (i, j) means cost of path from node i to node j


A
25
10
15
C
10
5
B
45
D