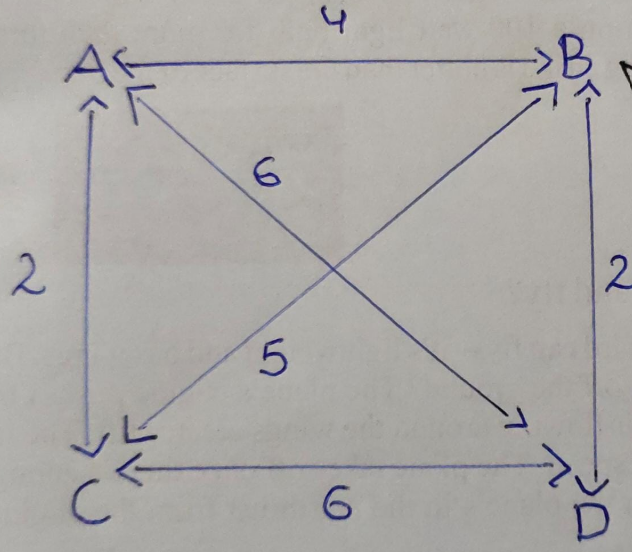
**Algorithm :**

1. **For Nearest neighbor heuristic :**
2. Select any starting city.
3. Find the nearest city to your current one and go there.
4. If there are cities not visited till now, repeat Step (ii). Else return to starting city.

Here is the example to understand the algorithm in a better way:



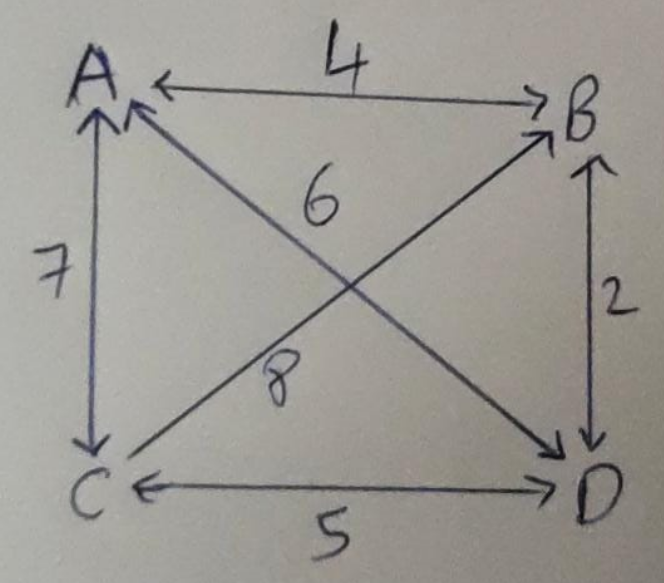
* Using the nearest neighbor algorithm on the above symmetric travelling salesman problem starting at city A, we would then travel to city B.
* We have to check all the possible paths from the current city and choose one under the condition that distance is minimum.
* From city B, we have two paths that is, B -> D and B -> C. Now, we have to choose the path which has shortest distance. So from city B we are travelling to city D. From city D, we have two paths that is, D -> A and D -> C. The shortest path between them D -> C. From C we have a path to return to starting path.
* This gives the total distance is 14.
* Suppose, at the start if we choose the path from A -> C which is 4. From C, we have two paths C -> B and C -> D but the shortest path is C -> B. So, we select this path. From B, travel to D as its distance is 2. From D, we have a path to return to starting path.
* This gives the total distance is 15, but this is not optimal. If we took the path A -> B -> D -> C -> A, the distance is 14. It is a little improvement in the algorithm. So, the algorithm achieved a sub-optimal route.

1. **For Cheapest link :**
2. Sort all paths in the network
3. Select the shortest path and add it if it does not violate the following rules:
4. There should be no cycles with less than n edges

(or)

1. Increase the degree of any city to more than 2
2. If we have n paths then repeat step (ii).

Here is the example to understand the algorithm in a better way:



* Sort the paths from the figure :

B <-> D = 2 , A <-> B = 4 , C <-> D = 5 , A <-> D = 6 , A <-> C = 7 , C -> B = 8

* Add B <-> D , A <-> B , C <-> D to the tour. We cannot add A <-> D as it creates a cycle and increases the order of D to 3. So, we skip this path and select A <-> C to the tour. This results the same solution which is obtained by nearest neighbor algorithm.
* This gives the total distance 18.