14. Traveling Salesman Problem (TSP)

Aim: To find the shortest Hamiltonian cycle that visits all cities exactly once using the Traveling Salesman Problem approach.

Algorithm (Brute Force / Dynamic Programming):

1. **Represent cities as a cost adjacency matrix.**
2. **Start from city 0.**
3. **Try all permutations of remaining cities.**
4. **Calculate path cost including returning to the start.**
5. **Keep track of the minimum cost path.**

```
#include <stdio.h>

#define INF 9999

#define N 4


int tsp(int graph[N][N], int mask, int pos, int dp[N][1<<N]) {

   if(mask==(1<<N)-1) return graph[pos][0];

   if(dp[pos][mask]!=-1) return dp[pos][mask];


   int ans=INF;

   for(int city=0;city<N;city++) {

      if(!(mask & (1<<city))) {

         int newAns=graph[pos][city]+tsp(graph,mask|(1<<city),city,dp);

         if(newAns<ans) ans=newAns;

      }

   }

   return dp[pos][mask]=ans;

}
```

```
int main() {
   int graph[N][N]={{0,10,15,20},{10,0,35,25},{15,35,0,30},{20,25,30,0}};
   int dp[N][1<<N];
   for(int i=0;i<N;i++) for(int j=0;j<(1<<N);j++) dp[i][j]=-1;
   printf("Minimum TSP cost: %d\n", tsp(graph,1,0,dp));
   return 0;
}
```

Sample Output:

Minimum TSP cost: 80

Result: The minimum traveling cost was found successfully using dynamic programming for TSP.