1. **Travelling salesman using dynamic programming**

**Code:**

#include <stdio.h>

#include <limits.h>

#define V 10

int min(int x, int y) { return (x < y) ? x : y; }

int tsp(int graph[][V], int mask, int pos, int n, int dp[][V]) {

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

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

int ans = INT\_MAX;

for (int city = 0; city < n; city++)

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

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

ans = min(ans, newAns);

}

return dp[mask][pos] = ans;

}

int main() {

int n;

printf("Enter the number of vertices (maximum %d): ", V);

scanf("%d", &n);

int graph[V][V], dp[1 << V][V];

printf("Enter the cost matrix (%d x %d):\n", n, n);

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++) scanf("%d", &graph[i][j]);

for (int i = 0; i < (1 << V); i++)

for (int j = 0; j < V; j++) dp[i][j] = -1;

int minCost = tsp(graph, 1, 0, n, dp);

printf("Minimum cost of the Hamiltonian cycle: %d\n", minCost);

return 0;

}

**Output:**

Enter the number of vertices (maximum 10): 4

Enter the cost matrix (4 x 4):

0 10 15 20

10 0 35 25

15 35 0 30

20 25 30 0

Minimum cost of the Hamiltonian cycle: 80

--------------------------------

Process exited after 23.85 seconds with return value 0

Press any key to continue . . .

