



FileEditSelectionViewGoRunTerminalHelp

tsp.cpp - DAA - Visual Studio Code

n_queens.cppbellman_ford.cpptsp.cpp

▶□□□

tsp.cpp > main()

```
38 void reduce_row(int matrix_reduced[N][N], int row[N])
39 {
40     fill_n(row, N, INF);
41     for (int i = 0; i < N; i++)
42     {
43         for (int j = 0; j < N; j++)
44         {
45             if (matrix_reduced[i][j] < row[i])
46             {
47                 row[i] = matrix_reduced[i][j];
48             }
49         }
50     }
51     for (int i = 0; i < N; i++)
52     {
53         for (int j = 0; j < N; j++)
54         {
55             if (matrix_reduced[i][j] != INF && row[i] != INF)
56             {
57                 matrix_reduced[i][j] -= row[i];
58             }
59         }
60     }
61 }
62
63 void reduce_column(int matrix_reduced[N][N], int col[N])
64 {
65     fill_n(col, N, INF);
66     for (int i = 0; i < N; i++)
67     {
68         for (int j = 0; j < N; j++)
69         {
70             if (matrix_reduced[i][j] < col[j])
71             {
72                 col[j] = matrix_reduced[i][j];
73             }
74         }
75     }
76     for (int i = 0; i < N; i++)
```

Ln 200, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

File Edit Selection View Go Run Terminal Help

tsp.cpp - DAA - Visual Studio Code

n_queens.cpp bellman_ford.cpp tsp.cpp

main()

```
76     for (int i = 0; i < N; i++)
77     {
78         for (int j = 0; j < N; j++)
79         {
80             if (matrix_reduced[i][j] != INF && col[j] != INF)
81             {
82                 matrix_reduced[i][j] -= col[j];
83             }
84         }
85     }
86 }
87
88 void display_matrix(int matrix_reduced[N][N])
89 {
90     for (int i = 0; i < N; i++)
91     {
92         for (int j = 0; j < N; j++)
93         {
94             if (matrix_reduced[i][j] == INF)
95             {
96                 cout << "∞ ";
97             }
98             else
99             {
100                 cout << matrix_reduced[i][j] << " ";
101             }
102         }
103         cout << "\n";
104     }
105 }
106
107 int compute_cost(int matrix_reduced[N][N])
108 {
109     int cost = 0;
110     int row[N];
111     reduce_row(matrix_reduced, row);
112     int col[N];
113     reduce_column(matrix_reduced, col);
114     for (int i = 0; i < N; i++)
115     {
```

Ln 200, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

File Edit Selection View Go Run Terminal Help tsp.cpp - DAA - Visual Studio Code

n_queens.cpp bellman_ford.cpp tsp.cpp

tsp.cpp > main()

```
107 int compute_cost(int matrix_reduced[N][N])
108 {
109     int cost = 0;
110     int row[N];
111     reduce_row(matrix_reduced, row);
112     int col[N];
113     reduce_column(matrix_reduced, col);
114     for (int i = 0; i < N; i++)
115     {
116         cost += (row[i] != INT_MAX) ? row[i] : 0;
117         cost += (col[i] != INT_MAX) ? col[i] : 0;
118     }
119     cout << "\nReduced Matrix :: \n\n";
120     display_matrix(matrix_reduced);
121     return cost;
122 }
123
124 void display_path(vector<pair<int, int>> const &list)
125 {
126     cout << "\n\nPath :: \n\n";
127     for (int i = 0; (unsigned)i < list.size(); i++)
128     {
129         cout << list[i].first + 1 << " -> " << list[i].second + 1 << endl;
130     }
131 }
132
133 struct comp
134 {
135     // comparator structure for node cost (operator override)
136     bool operator()(const Node *lhs, const Node *rhs) const
137     {
138         return lhs->cost > rhs->cost;
139     }
140 };
141
142 int solveTSP(int costMatrix[N][N])
143 {
144     priority_queue<Node *, vector<Node *>, comp> pq;
145     vector<pair<int, int>> v;
146     Node *root = new Node(costMatrix[0][0], 0, 1, 0);
```

Ln 200, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

File Edit Selection View Go Run Terminal Help

tsp.cpp - DAA - Visual Studio Code

n_queens.cpp bellman_ford.cpp tsp.cpp

tsp.cpp > main()

```
142 int solveTSP(int costMatrix[N][N])
143 {
144     priority_queue<Node *, vector<Node *>, comp> pq;
145     vector<pair<int, int>> v;
146     Node *root = newNode(costMatrix, v, 0, -1, 0);
147     root->cost = compute_cost(root->matrix_reduced);
148     pq.push(root);
149     while (!pq.empty())
150     {
151         Node *min = pq.top();
152         pq.pop();
153         int i = min->vertex;
154         if (min->level == N - 1)
155         {
156             min->path.push_back(make_pair(i, 0));
157             display_path(min->path);
158             return min->cost;
159         }
160         for (int j = 0; j < N; j++)
161         {
162             if (min->matrix_reduced[i][j] != INF)
163             {
164                 Node *child = newNode(min->matrix_reduced, min->path, min->level + 1, i, j);
165                 child->cost = min->cost + min->matrix_reduced[i][j] +
166                             compute_cost(child->matrix_reduced);
167                 pq.push(child);
168             }
169         }
170         delete min;
171     }
172     return 0;
173 }
174
175 int main()
176 {
177     int costMatrix[N][N], result;
178     cout << "Enter the cost matrix :: \n";
179
180     for (int i = 0; i < N; i++)
```

Ln 200, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

File Edit Selection View Go Run Terminal Help

tsp.cpp - DAA - Visual Studio Code

n_queens.cpp bellman_ford.cpp tsp.cpp

main()

return 0;

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

int main()

{

int costMatrix[N][N], result;

cout << "Enter the cost matrix :: \n";

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

{

for (int j = 0; j < N; j++)

{

if (i == j)

{

costMatrix[i][j] = INF;

}

else

{

cout << "Enter the cost of edge " << i + 1 << " -> " << j + 1 << " : ";

cin >> costMatrix[i][j];

}

}

}

result = solveTSP(costMatrix);

cout << "\n\nTotal Cost :: " << result << "\n\n";

return 0;

}

Ln 200, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

File Edit Selection View Go Run Terminal Help

tsp.cpp - DAA - Visual Studio Code

n_queens.cpp bellman_ford.cpp tsp.cpp

tsp.cpp > display_matrix(int [N][N])

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  #define N 4
6  #define INF INT_MAX
7
8  struct Node
9  {
10     vector<pair<int, int>> path;
11     int matrix_reduced[N][N];
12     int cost;
13     int vertex;
14     int level;
15 };
16
17 Node *newNode(int matrix_parent[N][N], vector<pair<int, int>> const &path, int level, int i, int j)
18 {
19     Node *node = new Node;
20     node->path = path;
21     if (level != 0)
22     {
23         node->path.push_back(make_pair(i, j));
24     }
25     memcpy(node->matrix_reduced, matrix_parent,
26           sizeof node->matrix_reduced);
27     for (int k = 0; level != 0 && k < N; k++)
28     {
29         node->matrix_reduced[i][k] = INF;
30         node->matrix_reduced[k][j] = INF;
31     }
32     node->matrix_reduced[j][0] = INF;
33     node->level = level;
34     node->vertex = j;
35     return node;
36 }
37
38 void reduce_row(int matrix_reduced[N][N], int row[N])
39 {
40     fill_n(row, N, INF);
```

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\DELL\Desktop\DAA> .\tsp.exe

Enter the cost matrix ::

Enter the cost of edge 1 -> 2 : 10

Enter the cost of edge 1 -> 3 : 15

Enter the cost of edge 1 -> 4 : 20

Enter the cost of edge 2 -> 1 : 10

Enter the cost of edge 2 -> 3 : 35

Enter the cost of edge 2 -> 4 : 25

Enter the cost of edge 3 -> 1 : 15

Enter the cost of edge 3 -> 2 : 35

Enter the cost of edge 3 -> 4 : 30

Enter the cost of edge 4 -> 1 : 20

Enter the cost of edge 4 -> 2 : 25

Enter the cost of edge 4 -> 3 : 30

Reduced Matrix ::

- 0 0 0

0 - 20 5

0 20 - 5

0 5 5 -

Reduced Matrix ::

- - -

- - 10 0

0 - - 5

0 - 0 -

Reduced Matrix ::

- - -

0 - - 5

- 10 - 0

0 0 - -

Reduced Matrix ::

- - -

0 - 20 -

0 20 - -

Ln 96, Col 27 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

