

Dijkstra's Algorithm:

```
*test.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

*test.cpp X test_tcs.cpp X
1 // Dijkstra's algorithm
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 void ShortestPaths(int v, vector<vector<int>>& cost, vector<int>& dist, int n) {
6     vector<bool> s(n, false);
7     dist.assign(n, INT_MAX);
8     dist[v] = 0;
9
10    for (int num = 0; num < n - 1; num++) {
11        int u = -1;
12        for (int i = 0; i < n; i++) {
13            if (!s[i] && (u == -1 || dist[i] < dist[u]))
14                u = i;
15        }
16        if (u == -1) break;
17        s[u] = true;
18
19        for (int w = 0; w < n; w++) {
20            if (!s[w] && cost[u][w] != INT_MAX && dist[u] + cost[u][w] < dist[w]) {
21                dist[w] = dist[u] + cost[u][w];
22            }
23        }
24    }
25 }
26
27 int main() {
28     vector<vector<int>> cost = {
29         {INT_MAX, 1, 3, INT_MAX, INT_MAX, INT_MAX, 10, INT_MAX},
30         {1, INT_MAX, 1, 7, 5, INT_MAX, 2},
31         {3, 1, INT_MAX, 9, 3, INT_MAX, INT_MAX},
32         {INT_MAX, 7, 9, INT_MAX, 2, 1, 12},
33         {INT_MAX, 5, 3, 2, INT_MAX, INT_MAX},
34         {10, INT_MAX, INT_MAX, 1, 2, INT_MAX, INT_MAX},
35         {INT_MAX, 2, INT_MAX, 12, INT_MAX, INT_MAX, INT_MAX}
36     };
37
38     int n = cost.size();
39     vector<int> dist;
40
41     ShortestPaths(0, cost, dist, n);
42
43     cout << "Shortest distances from node 0:" << endl;
44     for (int i = 0; i < n; i++) {
45         if (dist[i] == INT_MAX) {
46             cout << "Node 0 to Node " << i << ": Unreachable" << endl;
47         } else {
48             cout << "Node 0 to Node " << i << ": " << dist[i] << endl;
49         }
50     }
51
52     return 0;
53 }
```

Shortest distances from node 0:
Node 0 to Node 0: 0
Node 0 to Node 1: 1
Node 0 to Node 2: 2
Node 0 to Node 3: 7
Node 0 to Node 4: 5
Node 0 to Node 5: 7
Node 0 to Node 6: 3
Process returned 0 (0x0) execution time : 0.107 s
Press any key to continue.

```
"C:\Users\Golam Mostafa Rabby\OneDrive\B.Sc in Duet\3rd year 1st semeste...
Shortest distances from node 0:
Node 0 to Node 0: 0
Node 0 to Node 1: 1
Node 0 to Node 2: 2
Node 0 to Node 3: 7
Node 0 to Node 4: 5
Node 0 to Node 5: 7
Node 0 to Node 6: 3
Process returned 0 (0x0) execution time : 0.107 s
Press any key to continue.
```

test_lcs.cpp - Code::Blocks 20.03

test.cpp	X
test_lcs.cpp	X

```

1 // longest common subsequence
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 string findLCS(const string& a, const string& b)
6 {
7     int n = a.length(), m = b.length();
8     vector<vector<int>>> dp(n + 1, vector<int>(m + 1, 0));
9
10    for (int i = 1; i <= n; i++)
11    {
12        for (int j = 1; j <= m; j++)
13        {
14            if (a[i - 1] == b[j - 1])
15                dp[i][j] = 1 + dp[i - 1][j - 1];
16            else
17                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
18        }
19    }
20
21    for (int i = 0; i <= n; i++)
22    {
23        for (int j = 0; j <= m; j++)
24        {
25            cout << dp[i][j] << " ";
26        }
27        cout << endl;
28    }
29
30    string lcs = "";
31    int i = n, j = m;
32    while (i > 0 && j > 0)
33    {
34        if (a[i - 1] == b[j - 1])
35        {
36            lcs += a[i - 1];

```

```

31 int i = n, j = m;
32 while (i > 0 && j > 0)
33 {
34     if (a[i - 1] == b[j - 1])
35     {
36         lcs += a[i - 1];
37         i--;
38         j--;
39     }
40     else if (dp[i - 1][j] > dp[i][j - 1])
41     {
42         i--;
43     }
44     else
45     {
46         j--;
47     }
48 }
49 reverse(lcs.begin(), lcs.end());
50 return lcs;
51 }
52
53 int main()
54 {
55     string str1 = "lullabybabies";
56     string str2 = "skullandbones";
57     string lcs = findLCS(str1, str2);
58     cout << "Length: " << lcs.length() << endl;
59     cout << "LCS: " << lcs << endl;
60     return 0;
61 }

```

```
Process returned 0 (0x0)   execution time : 0.4
20 5
Press any key to continue.
```