

Dijkstra's Algorithm

Shikha-N

18 MIPES149
07-12-20

```
#include <bits/stdc++.h>
using namespace std;
int ar[30][30], source, d[30], p[30];
void dijkstraAlg (int ar[][30], int n) {
    int s[n];
```

```
    for (int i=0; i<n; i++) {
        d[i] = ar[source][i];
        p[i] = source;
        s[i] = 0;
```

```
    }
    s[source] = 1;
```

```
    for (int c=0; c<n; c++) {
        int min=999, u;
```

```
        for (int j=0; j<n; j++) {
```

```
            if (d[j] < min && s[j] != 1) {
```

```
                min = d[j];
```

```
                u = j;
```

```
            }
```

```
        s[u] = 1;
```

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

```
            if (min + ar[u][i] < d[i]) {
```

```
                d[i] = min + ar[u][i];
```

```
                p[i] = u;
```

```
            }
        }
    }
```

```
int main() {
```

```
    int n;
```

```
    cout << "Enter the no. of nodes vertices : " << endl;
```

```
    cin >> n;
```

```
    cout << "Enter the adjacency matrix (Enter 999  
    for infinity) : " << endl;
```

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

```
        for (int j = 0; j < n; j++) {
```

```
            cin >> a[i][j];
```

```
        }
```

```
    }
```

```
    cout << "Enter the Source router : " << endl;
```

```
    cin >> source;
```

```
    cout << "The shortest paths from source  
    router
```

```
    " : " << source << " are : " << endl;
```

```
    dijkstra(a, n);
```

```
return
```

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

```
        int k = i;
```

```
        while (k != source) {
```

```
            cout << k << " < ";
```

```
            k = p[k];
```

```
        } cout << source << " ";
```

```
        cout << d[i] << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

(2)

shikha

Shikha N
18M18CS149