

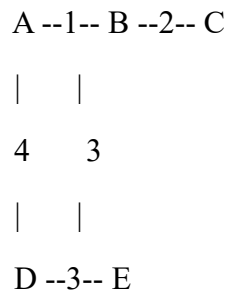
Dijkstra's Algorithm:

It acts as the heart of our project as it helps in finding the shortest path in Maps(Google Maps) and also in network optimization.

Core Idea of our Algorithm:

" Repeatedly picks the closest unvisited node and updates distances to its neighbors. "

Let's understand with an example:



If we want to find the shortest path based from A to C ,we have two ways but thanks to this algorithm we will get the path that has least weight.

Data Structures Used:

Graphs for storing neighbours and weights

Distance Table(can be Adjacency List) for calculating the shortest distance

Min-heap for picking the smallest distance node

Pseudocode :

```
void Graph::Dijkstra(int start) {
    vector<int> dist(V, INT_MAX);
    dist[start] = 0;
```

```

// Min-heap: (distance, node)

priority_queue<pair<int,int>, vector<pair<int,int>>, greater<pair<int,int>>> pq;

pq.push({0, start});

while(!pq.empty()) {
    int u = pq.top().second;
    int d = pq.top().first;
    pq.pop();

    // Skip if we already found a better distance
    if(d > dist[u]) continue;

    // Traverse neighbors
    for(auto [v, w] : adj[u]) {
        if(dist[u] + w < dist[v]) {
            dist[v] = dist[u] + w;
            pq.push({dist[v], v});
        }
    }
}

cout << "Shortest distances from node " << start << ":\n";
for(int i = 0; i < V; i++)
    cout << "Node " << i << ": " << dist[i] << "\n";
}

```

Time Complexity:  $O(V^2)$

Space Complexity:  $O(V+E)$

Advantages of using this Algorithm:

Helps in navigation like GPS

For Logistics Planning

Computer Networks

Resource Allocation