10/25/2024

DISCRETE STRUCTURES

Show N Tell

Muhammad Ali Amir

BSCS24137

Instructor : Muddasir Shabbir

Show N Tell Topic :

PATH FINDING

Sub Topic :

Dijkstra algorithm

Introduction :

Dijkstra’s Algorithm was discovered by the Dutch computer scientist Edsger W. Dijkstra in 1956. It is a graph search algorithm used to find the shortest path between nodes in a graph, which may represent, for example, road networks. The algorithm is particularly efficient in graphs with non-negative weights and is commonly used in real-life applications such as GPS navigation, social networking, and networking systems like telephone networks.

Explanation :

* Initialization: Assign to every node a tentative distance. Set the initial node’s distance to 0 and all other nodes’ distances to infinity. Mark all nodes as unvisited. The initial node becomes the current node.
* Relaxation: For each unvisited neighbour of the current node, calculate the tentative distance through the current node. If this distance is smaller than the current assigned distance of that neighbour, update the tentative distance.
* Mark visited: Once all neighbours of the current node have been considered, mark the current node as visited. A visited node will not be checked again.
* Selection of the next node: Select the unvisited node with the smallest tentative distance, and make it the new current node.
* Repeat: Repeat steps 2–4 until the destination node has been visited or all possible nodes have been processed.

Relaxation Formula :

Dijkstra’s Algorithm uses relaxation formula for identification of shortest path,which is given as :

If

D(V)>D(U) + C(U,V)

Then

D(V)=D(U)+C(U,V)

Example :

• You have a graph with nodes labeled 1 to 6.

• The algorithm starts at node 1, and its goal is to find the shortest path to other nodes.

Here’s how the relaxation happens:

• From node 1 to node 2: the distance D(2) becomes 7.

• From node 2 to node 3: the distance D3) becomes 9.

• From node 2 to node 6: the distance D(6) becomes 14.

The relaxation formula:

If

D(V)>D(U) + C(U,V)

Then

D(V)=D(U)+C(U,V)

• D(U) is the distance to the current node.

• C(U,V) is the cost between node U and V.

• D(V) is the distance to the neighbour node.

This process continues until the shortest path to each node is found.

Real Life Applications :

* Google Maps
* DNA Mapping
* Social Networking
* Telephone Networks