

Dijkstra's Algorithm

Crishank
4/10/17
AZ

- An algorithm for finding the shortest paths b/w nodes in a graph - conceived by Edsger W. Dijkstra 1956
- Multiply Variants

Given a graph & a source vertex in a graph, find shortest paths from source to all vertices in the given graph

- Original - shortest path b/w nodes
- common: single node fixed as the "source" node & finds shortest paths from the source to all other nodes in graph
 - shortest path tree
- Shortest path algorithms - used for routing protocols etc.
- Uniform-cost search - variant of BFS Alg used especially in artificial intelligence
- formulated as an instance of the more general idea of best-first search (greedy alg)

ALGORITHM - assigns some initial distance values & tries to improve them step by step

1. Assign to every node a tentative distance value
 - set it to zero for initial node, & ∞ for all other nodes
2. Set the initial node as current, all other nodes = unvisited; create set of all unvisited nodes = unvisited set
3. Current node, calculate tentative distances to all unvisited neighbors, compare newly calculated tent. dist. to current assigned val & assign smaller one

Eg. - current node A = distance 0

- edge connecting w/ neighbor B = length 2

- So dist to B (through A) is $0 + 2 = 2$

- IF B was previously marked w/ a dist > 2 , change to 2. ELSE, keep current val

4. When done w/ considering current node's neighbor, mark current node as visited & rem from unvisited set
 - a visited node will never be checked again

5. STOP if the destination node = visited OR if smallest tentative distance among unvisited set = ∞ . ALGO = DONE

6. OTHERWISE select unvisited node w/ smallest tent. dist. (= current node) step 3

loop from
6 to 3 until
5's condition is true

 ∞ occurs when there is no connection b/w start & remaining unvisited nodes