On Dijkstra's Lemma and Kruskal's Algorithm

Dijkstra's Lemma Consider an undirected graph that consists of a single cycle and in which each vertex is marked to be either an A-node or a B-node. Then the number of AB-edges - i.e. edges that connect an A-node with a B-node - is even.

Let, for a given finite set of vertices, for each pair the length of the edge between them be given; for simplicity's sake all edge lengths are assumed to be distinct. We now give two theorems about "the shortest tree", i.e. the spanning tree with the smallest length, where the length of a tree is defined as the sum of the lengths of its edges.

Theorem 0. The longest edge of a cyclic path does not belong to the shortest tree.

Proof Let e be the longest edge of a cyclic path. We prove Theorem 0 by showing that for any spanning tree T that contains e we can construct a shorter spanning tree. Remove edge e from tree T, which thus falls apart into two subtrees which we call A and B respectively; e being an AB-edge in the cyclic path, Dijkstra's Lemma tells us that the cyclic path contains another AB-edge, which we use to reconnect A and B. The resulting tree is shorter than T because e is the longest edge of the cyclic path. (End of Proof.)

Theorem 1 Partition the vertices into A-nodes and B-nodes. Then the shortest AB-edge belongs to the shortest tree.

Proof Let, for a given partitioning, e be the shortest AB-edge. We prove Theorem 1 by showing that for any spanning tree T that does not contain e, we can construct a shorter spanning tree. Add edge e to tree T, thus creating 1 cyclic path, which contains e; e being an AB-edge, Dijkstra's Lemma tells us that that cyclic path contains another AB-edge, which we remove. The resulting tree is shorter because e is the shortest AB-edge. (End of Proof.)

Kruskal's Algorithm constructs the shortest tree by processing - i.e. "rejecting" or "accepting" - the edges in the order of increasing length. At any stage, the accepted edges construct a forest - to begin with as many isolated roots as there are vertices, and finally 1 big tree of accepted edges - . Kruskal's Algorithm rejects a next edge if it forms a cycle with (some of) the edges that have already been accepted, otherwise it accepts it. In terms of the forest: the next edge is rejected if it connects two nodes from the same tree, and accepted if it connects two trees. Rejection is justified by Theorem 0, applied to the cyclic path formed in the tree; acception is justified by Theorem 1 if we partition the forest into A-trees and B-trees such that the edge under consideration is an AB-edge.

Austin, 3 May 1998

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
USA