MA2C03 Mathematics
School of Mathematics, Trinity College
Hilary Term 2016
Lecture 35 (January 27, 2016)

David R. Wilkins

34. Kruskal's Algorithm

Definition

Let (V, E) be an undirected graph whose set of vertices is V and whose set of edges is E. A *cost* function $c\colon E \to \mathbb{R}$ on the set E of edges of the graph is a function that assigns to each edge e of the graph a real number $c(e)$.

Let $c\colon E \to \mathbb{R}$ be a cost function on the set E of edges of a graph (V, E). Given any subset S of E, we define

$$c(S) = \sum_{e \in S} c(e).$$

(Thus $c(S)$ is the sum of the costs of all the edges of the graph that belong to the subset E.)

Let (V, E) be a connected graph. We recall that a subgraph of (V, E) is said to be a *spanning tree* if it is a tree that includes all the vertices of the given connected graph. Thus a subgraph of that given graph is a spanning tree if and only if it is a connected acyclic subgraph of the given graph that includes all the vertices of the given graph.

Definition

Let (V, E) be a connected graph on which is defined a cost function $c\colon E \to \mathbb{R}$ that assigns a cost $c(e)$ to each edge e of the graph. A spanning tree (V_M, E_M) is said to be *minimal* (with respect to this cost function) if $c(E_M) \leq c(E_T)$ for all spanning trees (V_T, E_T) of the given graph.

We discuss *Kruskal's Algorithm* for finding minimal spanning trees. Let (V, E) be a graph, and let $c\colon E \to \mathbb{R}$ be a cost function defined on the set E of edges of the graph (V, E). We start with a subgraph that initially consists of all the vertices of the graph (with no edges). List the edges of the graph in a *queue* so that the the edges is non-decreasing in the queue. (Thus if e and e' are edges of the graph, and if $c(e) < c(e')$ then e precedes e' in the queue.) Take edges successively from the front of the queue, and determine whether or not addition of that edge to the current subgraph will create a cycle. If such a cycle would be created then discard it; otherwise add it to the subgraph. Continue till the queue is emptied.

The algorithm described always yields a minimal spanning tree for the given graph.
We now justify this assertion.
Let the edges in the queue that are added to the subgraph as that subgraph is built up be ordered in sequence as $e_1, e_2, e_3, \ldots, e_m$. The spanning tree ultimately constructed then consists of the vertices of the original graph, together with the edges $e_1, e_2, \ldots, e_m$. Moreover $c(e_i) \leq c(e_j)$ whenever $i < j$. Also if e' is an edge of the original graph, if $c(e') \leq c(e_k)$ for some integer j betweeen 2 and m, and if the subgraph consisting of the $e_1, e_2, \ldots, e_{m-1}$ and e' together with the endpoints of those edges would contain a cycle, then the edge e' is disarded by the Kruskal algorithm, and is thus not included in the spanning tree constructed.