**Dijkstra: Introduction**

**Finding distance in weighted graphs** (This topic is in the course reader.)

In the section on Steiner trees, it was necessary to find the distance graph $D_G(N)$, which is determined by the distances between the vertices in $N$.

In 1959, Dijkstra gave an algorithm for finding distances from a vertex $u_0$ to all other vertices in a weighted graph. In fact, it gives a tree with root vertex $u_0$, such that the tree contains shortest paths from $u_0$ to all other vertices in the graph.

At intermediate stages of the algorithm, there is the following structure in the graph

When this structure is known, we can choose the vertex $v \in \bar{S}$ with lowest label: there can be no shorter path to any vertex in $\bar{S}$. Hence $d(u_0, v) = \ell(v)$. Thus we can update by adding $v$ to $S$.

This leads to the following:

**Dijkstra’s Algorithm** (determines distance from $u_0$ to all other vertices in a weighted graph $G$, along with a tree determining shortest paths from $u_0$)

1. Set $\ell(u_0) := 0, \ell(v) := \infty$ for all $v \neq u_0$,
   $i := 0, S := \{u_0\}, \bar{S} := V(G)\{u_0\}$

2. while $i < |V(G)| - 1$ do the following: (i.e do it $|V(G)| - 1$ times)
   (a) (relabeling vertices in $\bar{S}$)
      For each edges $u_i v$ with $v \in \bar{S}$,
      if $\ell(v) > \ell(u_i) + w(u_i v)$, set $\ell(v) := \ell(u_i) + w(u_i v)$
      and PARENT($v$) = $u_i$
      (this makes $v$ have a label of the length of the shortest path via $u_i$, if that is shorter than all paths previously found).
   (b) (choose the next $u_{i+1}$)
      Choose $v \in \bar{S}$ minimising $\ell(v)$, put $u_{i+1} := v$
   (c) $S := S \cup \{u_{i+1}\}, \bar{S} := \bar{S}\{u_{i+1}\}$

3. Output $\ell(v)$ for all $v$ (distance from $u_0$)
   and PARENT($v$) for all $v$ (predecessor in a shortest path from $u_0$ to $v$)
**Complexity**  Suppose $G$ has order $n$ with $m$ edges. Step 1 is $O(n)$. Step 2(a) examines each edge exactly once throughout the whole algorithm. Thus it takes $O(m)$ altogether. Step 2(b) requires, each time finding the smallest label in a set of $\leq n$ labels. This is $O(n)$ each time it is performed. It is performed $n$ times. So the complexity is $O(m) + O(n^2) = O(n^2)$ since $m \leq n(n-1)/2 < n^2$. 