Note Title 2/17/2010

CS 180 - Shortest paths in a graph

Announcements

- Posted review session info & worksheet
- Final exam next Monday at (10? noon?)

Shortest paths in a graph. (Ch 12)

Suppose we have $G=(V,E)$ and each edge $e \in E$ has a length l_e.

Here, we'll assume G is directed: $u \xrightarrow[l_{uv}]{10} v$.

Goal: Given two vertices, find shortest path between them.

(shortest path tree)

We'll actually do something harder:

Given a source vertex s, compute shortest path from s to every other vertex.

Greedy idea:

Start with a set S. ← set of vertices where I "know" the shortest path to s

(initially $S = \{s\}$)

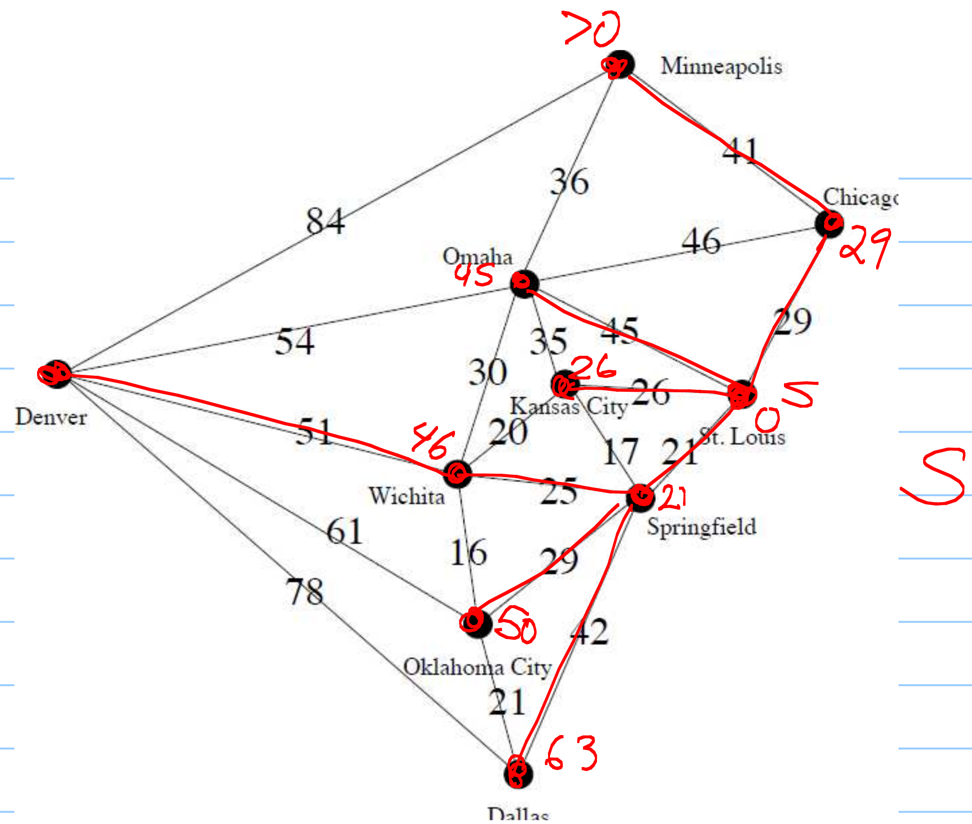
At each step, grow out from s, taking next shortest path from s to a new vertex & adding that to S.

Greedy idea:

Start with source (here, St. Louis)

Let $S = \{s\}$

Consider edges going out from S.



At each step, grow out from S, taking next shortest path from s to a new vertex & adding that to S.

Pseudo code: Dijkstra's algorithm

(actually Leyzorek, Gray, Johnson, Ladew, Meaker, Petry + Seitz)

SPtree(G, s):

$S \leftarrow \{s\}$

$D[s] \leftarrow 0$ ← distance array, initialized to ∞

$T \leftarrow \phi$

add red # to edges

while $S \neq V$

Select node v with at least one edge into S where $d'(v) = \min_{(u,v) \in E, u \in S} D[u] + \ell_{uv}$ is minimized

$S \leftarrow S \cup \{v\}$

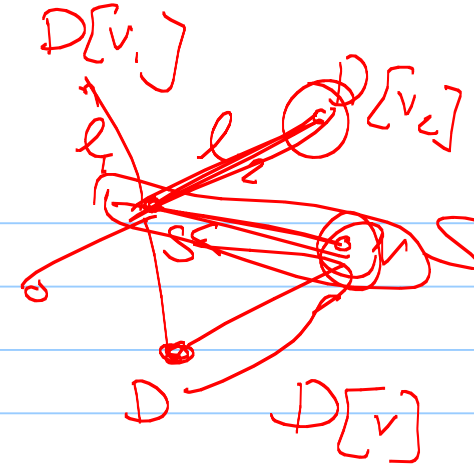
$D[v] \leftarrow d'(v)$

$T \leftarrow T \cup \{(u,v)\}$

Claim: At each stage, T is a set of shortest paths from s to S.

pf: induction on $|S|$

(go take 314)



Improved Pseudocode

Dijkstra(G, s):

Create array D[v], initially all ∞

$S \leftarrow \{s\}$

$D[s] \leftarrow 0$

for every edge (s, u)

  set $D[u] \leftarrow \ell_{su}$

While $S \neq V$

how long? $O(\log n)$ → select node $v \notin S$ with $D[v]$ minimized

  $S \leftarrow S \cup \{v\}$

  for each edge (v, u)

    if $D[v] + \ell(u,v) < D[u]$

      $D[u] \leftarrow D[v] + \ell(u,v)$

      $O(\log n)$

Runtime

$\leq d(v)$ times for each vertex's value $p[v]$ to be modified

$O(\log n)$ time each time

$$\sum_{v \in V} d(v) \underbrace{\log n} = \log n \underbrace{\sum_{v \in V} d(v)}$$

$$= (\log n)(2m)$$

$$\approx O(m \log n)$$