CSE 101 – Dec 2, 2019 (Week 10)

Notes provided by Ben Sihota bsihota@ucsc.edu

Lemma 2

After Initialize(G, s), the inequality:

$$\delta(s, x) \leq d[x] \qquad (\forall x \in V(G))$$

is maintained over **any** sequence of calls to Relax(.,.)

Proof (contradiction) Assume, to get a contradiction, that for some $x \in V(G)$ and after some relaxation sequence, we have

$$d[x] < \delta(s, x)$$

Then $d[x]$ **must** be finite. By Lemma 1, G contains an s˘x path of weight $d[x]$. This contradicts the very definition of $\delta(s, x)$ as the weight of a minimum weight s˘x path in G.

This contradiction shows our assumption was false, so no such $x \in V(G)$ and no such relaxation sequence exists. □

Lemma 3 (Path relaxation property)

If

$$P : s = x_0, x_1, x_2, ..., x_k$$

is a min weight s˘x_k path, and if edges of P are relaxed in order:

$$(x_0, x_1), (x_1, x_2), ..., (x_{k-1}, x_k)$$

then $d[x_k] = \delta(s, x_k)$.

Further, this remains true regardless of any other relaxations that may occur.

(See Lemma 24.15 on P. 673 for proof)

Bellman-Ford

Pseudocode for BellmanFord(G, s)

Total cost: $\Theta(mn)$ where $n = |V|$ and $m = |E|$

New ADT: Priority Queue

Maintains a finite set of elements (records) each with an associated 'key' attribubte

$$x = (.,\ .,\ .,\ .,\ \text{key}) \in S$$

(other stuff is satellite data)

$S = \{.,\ .,\ x,\ .,\ .\}$

S is a state in the ADT

ADT ops for a (max) priority queue

Operations

- Insert(S, x) – Inserts a new record into S
- Max(S) – Returns and deletes element with largest key
- IncreaseKey(S, x, k) – Changes $key[x]$ to k if $k > key[x]$, do nothing otherwise

Dijkstra

Pseudocode for Dijkstra(G, s)