Shortest paths

$G=(V,E)$, directed or undirected

$w:E\to\mathbf{R}$

path : $v_0, v_1, ..., v_k, (v_i, v_{i+1}) \in E$

$$w(p)=\sum_{i=0}^{k-1} w(v_i, v_{i+1})$$

p is a shortest path from v_0 to v_k if $w(p)\le w(p')$ for all paths p' from v_0 to v_k.

$D(u,v)=\infty$ if there is no path from v_0 to v_k

$D(u,v)=w(p)$, if there is a shortest path p from v_0 to v_k

$D(u,v)=-\infty$ otherwise

That is,

$D(u,v)=\min\{w(p)\mid p$ is a path from v_0 to $v_k\}$

where $w(p):=\infty$ if p is the empty path from v_0 to v_k.

Single-source : Given $s\in V$, to compute $D(s,v)\,\forall v\in V$.

All-pair : To compute $D(u,v)\,\forall u,v\in V$.

Dijkstra's algorithm for SSSD

$G = (V, E)$ undirected with $w(e) \geq 0 \forall e \in E$

$w(p) \geq 0$ for every path. $\Rightarrow$ removing cycles from a path cannot increase the weight of the path. $\Rightarrow$ For all $u, v \in V, u \neq v, \exists$ shortest path from u to v which is simple.

Obs: Suppose $w(s, v)$ is minimum for all $v \in V, v \neq s$.

Then $D(s, v) = w(s, v)$.

Moreover, if $(v, u) \in E$, then $D(s, v) + w(v, u)$ is the shortest distance of a path from s to u which lies within the set $\{s, v\}$ except for the last step.

Inductively, suppose

$T \subset V : \forall v \in T, D(s,v) = w(p)$ for some path that lies in T.

For $u \notin T$, let

$D_T(s,u) =$ shortest possible distance of a path from s to u which lies within T except for the last step.

$$D_T(s,u) = \min\{D(s,v) + w(v,u) \mid v \in T, (v,u) \in E\}$$

Lemma

$D_T(s,u) = D(s,u)$ if $D_T(s,u) \le D_T(s,w) \forall w \notin T$.

Pr*oof* : Let p be a path from s to v.

$p : s \xrightarrow{*} x \to y \xrightarrow{*} v$ where the subpath $s \xrightarrow{*} x$ lies in T and $y \notin T$.

$$w(p) \ge w(s \xrightarrow{*} x) + w(x,y) \ge D_T(s,x) + w(x,y)$$
$$\ge D_T(s,v).$$

Algorithm

$Q \leftarrow V; [Q = V - T]$

$D(s) \leftarrow 0; D(v) \leftarrow \infty \forall v \neq s.$

While Q is non-empty begin

$\quad v \leftarrow$ Extract-min(Q) w.r.t. D;

$\quad$ For every u adj. to v, if $u \in Q$ then

$\quad\quad D(u) \leftarrow \min\{D(u), D(v) + w(v,u)\};$

end

Time complexity:

Q can be maintained by a Fibonacci heap

O(n) extract-min $\Rightarrow$ O(nlogn)

O(e) comparisons

Total O(e + nlogn)

Bellman-Ford – source, negative wt. allowed

$\delta_k(v) :=$ the distance of shortest possible path from s to v with $\leq k$ edges.

$\delta_0(v) = \infty$ if $v \neq s$; $\delta_0(s) = 0$

$\delta_{k+1}(v) = \min\{\delta_k(v), \{\delta_k(u) + w(u,v) \mid (u,v) \in E\}$

Do the following $n-1$ times:

$\forall (u,v) \in E, \delta(v) \leftarrow \min\{\delta(v), \delta(u) + w(u,v)\};$

If $\forall (u,v) \in E, \delta(v) \leq \delta(u) + w(u,v)$ then done

otherwise a negative cycle is detected.

Analysis

$$\text{Let } \delta(v) = \delta_{n-1}(v)$$

Suppose G has no negative cycle.

$$\text{Then } \delta\ (v) = D(s, v).$$

$$\because \exists \text{ shortest simple path } s \xrightarrow{*} v.$$

$$\text{Hence, } \delta(v) \le \delta(u) + w(u, v) \forall (u, v) \in E$$

Suppose G has a negative cycle reachable from the source:

$C: v_0, v_1, \ldots, v_k, v_{k+1} = v_0$

Then $w(C) = \sum_{i=0}^{k} w(v_i, v_{i+1}) < 0$ and $\delta(v_i)$ is finite for $i = 0, \ldots, k$..

Claim: $\exists i, \delta(v_{i+1}) > \delta(v_i) + w(v_i, v_{i+1})$.

Pr*oof*: Otherwise $\exists i, \delta(v_{i+1}) \leq \delta(v_i) + w(v_i, v_{i+1})$.

$$\Downarrow$$

$$\sum_{i=0}^{k} \delta(v_{i+1}) \leq \sum_{i=0}^{k} \delta(v_i) + w(v_i, v_{i+1}).$$

$$\Downarrow$$

$$0 \leq \sum_{i=0}^{k} w(v_i, v_{i+1}).$$

A contradiction.

Floyd-Warshall -- all pair

Assume nongative weight.

$V = \{1, 2, ..., \mathrm{n}\}$

Path $p : v_1, \underbrace{v_2, ..., v_{k-1},}_{\text{intermediate nodes}} v_k$

$$D^{(k)}(i, j) := \min\{w(p) \mid p \text{ is a path } i \xrightarrow{*} j \text{ with intermediate nodes} \in \{1, ..., \mathrm{k}\}\ \}$$

$$D^{(k)}(i, j) = \min\{D^{(k-1)}(i, j), D^{(k-1)}(i, k) + D^{(k-1)}(k, j)\}$$

$\because$ Either $p : i \underbrace{\xrightarrow{*}}_{\text{all} \le k-1} j$

or $p : i \underbrace{\xrightarrow{*}}_{\text{all} \le k-1} k \underbrace{\xrightarrow{*}}_{\text{all} \le k-1} j$

Johnson’s all-pair

Either detects a neg. cycle or computes D(u,v)’s

Reduces to case of non-neg. wt by reweighting.

Uses Dijkstra’s.

Uses Bellman-Ford to detect neg. cycle.

Detecting neg. cycle using Bellman-Ford

Add a pseudo-source s and edges

$$(s, v) \forall v \in V.$$

$$w(s, v) \leftarrow 0.$$

G has a neg. cycle $\Leftrightarrow G'$ has a neg. cycle reachable from s.

$$\Downarrow$$

Run Bellman-Ford with source s:

Either a neg. cycle is detected,

or $\delta(\mathrm{v}) = D_{G'}(s, v)$ is computed for all v.

Note: $\delta(v) \le \delta(u) + w(u,v) \forall (u, v) \in E.$

Reweighting trick

Assign wt. on vertice s, $h : V \to \mathbf{R}$

$\hat{w}(u,v) := w(u,v) + h(u) - h(v)$.

$p : v_1, \ldots, v_k$, a path.

$$\hat{w}(p) = \sum_{i=1}^{k-1} \hat{w}(v_i, v_{i+1}) = \sum_{i=1}^{k-1} w(v_i, v_{i+1}) + h(v_{i+1}) - h(v_i)$$

$$= \sum_{i=1}^{k-1} w(v_i, v_{i+1}) + \sum_{i=0}^{k-1} h(v_{i+1}) - h(v_i)$$

$$= w(p) + h(v_0) - h(v_k)$$

$$\Downarrow$$

$p : v \xrightarrow{*} u$ is shortest wrt $w \Leftrightarrow p$ is shortest wrt $\hat{w}$.

Remark : If $v_0 = v_k$ (a cycle), then $\hat{w}(p) = w(p)$.

So, G has a neg. cycle $\Leftrightarrow G'$ has a neg. cycle.

Need $\hat{w}(u,v) = w(u,v) + h(u) - h(v) \geq 0 \forall (u,v) \in E$.

That is, $h(v) \leq h(u) + w(u,v)$.

Set $h(v) = \delta(v)$.

Bellman-Ford once -- O(ne)

Dijkstra from every node --O($n^2 \log n + ne$)