Algorithms: Design and Analysis, Part II

All-Pairs Shortest Paths (APSP)

A Reweighting Technique

Motivation

Recall: APSP reduces to n invocations of SSSP.

- nonnegative edge lengths: $O(mn \log n)$ via Dijkstra
- general edge lengths: $O(mn^2)$ via Bellman-Ford

as good as with nonnegative edge lengths!

Johnson's algorithm: Reduces APSP to

- 1 invocation of Bellman-Ford $(O(mn))$
- n invocations of Dijkstra $(O(nm \log n))$

Running time: $O(mn) + O(mn \log n) = O(mn \log n)$.

Tim Roughgarden

Quiz

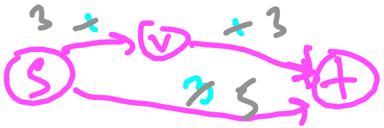
Suppose: $G=(V,E)$ directed graph with edge lengths. Obtain G' from G by adding a constant M to every edge's length. When is the shortest path between a source s and destination t guaranteed to be the same in G and G'?

(A) When G has no negative-cost cycle.

(B) When all edge costs of G are nonnegative.

(C) When all s-t paths in G have the same number of edges

(D) Always.



[M=2]

Quiz

Setup: $G=(V,E)$ is a directed graph with general edge lengths c_e. Fix a real number p_v for each vertex $v \in V$.

Definition: for every edge $e=(u,v)$ of G, $c'_e := c_e + p_u - p_v$

$$u \xrightarrow{c_e = 2} v$$

$p_u = -4$ $\quad$ $p_v = -3$

$c'_e = 2 + (-4) - (-3) = 1$

Question: If the s-t path P has length L with the original edge lengths $\{c_e\}$, what is P's length with the new edge lengths $\{c'_e\}$?

(A) L

(B) $L + p_s + p_t$

(C) $L + p_s - p_t$

(D) $L - p_s + p_t$

new length $= \sum_{e \in P} c'_e = \left(\sum_{e \in P} c_e\right) + p_s - p_t$

$= \sum_{e=(u,v) \in P} [c_e + p_u - p_v]$

Reweighting

Summary: reweighting using vertex weights $\{p_v\}$ adds the same amount (namely, $(p_s - p_t)$) to every s-t path

Consequence: reweighting always leaves the shortest path unchanged.

Why useful?: what if:

(1) G has some negative edge lengths → requires Bellman-Ford

(2) after reweighting by some $\{p_v\}$, all edge lengths become nonnegative! → enables Dijkstra!

Question: do such weights always exist?

- yes, and can be computed using the Bellman-Ford algorithm!