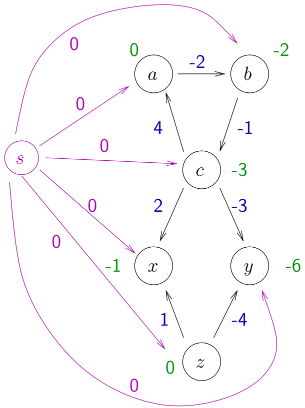
All-Pairs Shortest Paths (APSP)

Johnson's Algorithm

Algorithms: Design and Analysis, Part II

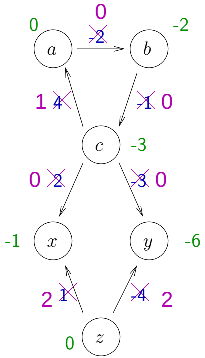
Example



Note: Adding s does not add any new u-v paths for any $u, v \in G$.

Key insight: Define vertex weight $p_v :=$ length of a shortest s-v path.

Example (con'd)



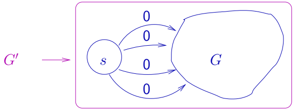
Recall: For each edge $e = (u, v)$, define $c'_e = c_e + p_u - p_v$.

Note: After reweighting, all edge lengths nonnegative! $\Rightarrow$ Can compute all (reweighted) shortest paths via n Dijkstra computations! [No need for Bellman-Ford]

Johnson's Algorithm

Input: Directed graph $G = (V, E)$, general edge lengths c_e.

(1) Form G' by adding a new vertex s and a new edge (s, v) with length 0 for each $v \in G$.



(2) Run Bellman-Ford on G' with source vertex s. [If B-F detects a negative-cost cycle in G' (which must lie in G), halt + report this.]

(3) For each $v \in G$, define p_v = length of a shortest $s \rightarrow v$ path in G'. For each edge $e = (u, v) \in G$, define $c'_e = c_e + p_u - p_v$.

(4) For each vertex u of G: Run Dijkstra's algorithm in G, with edge lengths $\{c'_e\}$, with source vertex u, to compute the shortest-path distance $d'(u, v)$ for each $v \in G$.

(5) For each pair $u, v \in G$, return the shortest-path distance $d(u, v) := d'(u, v) - p_u + p_v$

Analysis of Johnson's Algorithm

Running time: $O(n) + O(mn) + O(m) + O(nm \log n) + O(n^2)$

Step (1), form G' Step (2), run BF Step (3), form c' Step (4), n Dijkstra Step (5), $O(1)$ work per u-v pair

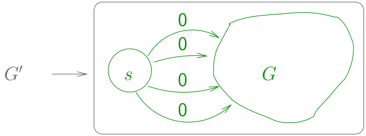
$= O(mn \log n)$. [Much better than Floyd-Warshall for sparse graphs!]

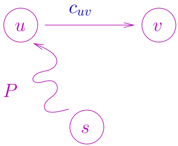
Correctness: Assuming $c'_e \geq 0$ for all edges e (see next slide for proof), correctness follows from last video's quiz.

[Reweighting doesn't change the shortest u-v path, it just adds $(p_u - p_v)$ to its length]

Correctness of Johnson's Algorithm

Claim: For every edge $e = (u, v)$ of G, the reweighted length $c'_e = c_e + p_u - p_v$ is nonnegative.





Proof: Fix an edge (u, v). By construction,
p_u = length of a shortest s-u path in G'
p_v = length of a shortest s-v path in G'
Let P = a shortest s-u path in G' (with length p_u - exists, by construction of G')
$\Rightarrow P + (u, v)$ = an s-v path with length $p_u + c_{uv}$
$\Rightarrow$ Shortest s-v path only shorter, so $p_v \leq p_u + c_{uv}$
$\Rightarrow c'_{uv} = c_{uv} + p_u - p_v \geq 0$. QED!