All-Pairs Shortest Paths (APSP)

Algorithms: Design and Analysis, Part II

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

Johnson's algorithm: Reduces APSP to

- 1 invocation of Bellman-Ford ($O(mn)$)

- n invocations of Dijkstra ($O(nm \log n)$)

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

As good as with nonnegative edge lengths!

Quiz

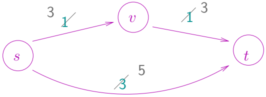
Suppose: $G = (V, E)$ directed graph with edge lengths. Obtain G' from G by adding a constant M to every edge's length. When is the shortest path between a source s and a destination t guaranteed to be the same in G and G'?

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

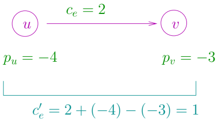


Quiz

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, $c'_e := c_e + p_u - p_v$

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 length $\{c'_e\}$?



A) L

B) $L + p_s + p_t$

C) $\boxed{L + p_s - p_t}$

D) $L - p_S + p_t$

New length $= \sum_{e \in P} c'_e = \sum_{e=(u,v) \in P} [c_e + p_u - p_v] = \left(\sum_{e \in P} c_e\right) + p_s - p_t$

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

(2) After reweighting by some $\{p_v\}$, all edge lengths become nonnegative!

Question: Do such weights always exist?

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

Requires Bellman-Ford, enables Dijkstra!