ALL-PAIRS SHORTEST PATHS

Input: weighted graph
- assume no negative cycles

Output: min-weight path for every pair of vertices

How would you solve this?
Intuitive solution: run SSSP, \( V \) times (once per vertex-source)

Cost:
\[
\begin{align*}
\text{V} \cdot O(V \cdot E) & \quad \text{w/ B-F} \\
\text{V} \cdot O(E \log V) & \quad \text{w/ bin.heap Dijkstra (adj. list)} \\
\text{V} \cdot \Theta(V^2) & \quad \text{w/ "array" Dijkstra (adj. matrix)} \\
\text{V} \cdot O(E + V \log V) & \quad \text{w/ Fibonacci heap Dijkstra (adj. list)}
\end{align*}
\]

require weights \( \geq 0 \)

To handle negative weights, all we get is \( O(V^2 E) = O(V^4) \)
JOHNSON'S ALGORITHM

\[ 1 \text{ Bellman-Ford } + \text{ ADJUST WEIGHTS } + V \cdot \text{Dijkstra} \]

- need weights \( \geq 0 \)
- need same shortest path structure

assume adj. list
Input weights: $w_{ij}$
For every vertex $k \ (1 \leq k \leq V)$ let $r_k$ be some real number.

Input weights: $w_{ij}$
For every vertex $k$ ($1 \leq k \leq V$) let $r_k$ be some real number.

Define $w'_{ij} = w_{ij} + r_i - r_j$
For every vertex $k$ ($1 \leq k \leq V$), let $r_k$ be some real number.

Define $w'_{ij} = w_{ij} + r_i - r_j$

Suppose path $p_1(a \rightarrow b)$ is better than path $p_2(a \rightarrow b)$

$\text{e.g.: } w_{14} + w_{43} + w_{36} < w_{15} + w_{56}$

Input weights: $w_{ij}$
For every vertex \( k \) \((1 \leq k \leq V)\) let \( r_k \) be some real number.

Define \[ w'_{ij} = w_{ij} + r_i - r_j \]

Suppose path \( p_1(a \rightarrow b) \) is better than path \( p_2(a \rightarrow b) \)

\[ \text{e.g.: } w_{14} + w_{43} + w_{36} < w_{15} + w_{56} \]

Compare paths with new weights:

\[ (w_{14} + r_1 - r_4) + (w_{43} + r_4 - r_3) + (w_{36} + r_3 - r_6) \text{ vs } (w_{15} + r_1 - r_5) + (w_{56} + r_5 - r_6) \]
For every vertex $k$ ($1 \leq k \leq V$) let $r_k$ be some real number.

Define $w'_{ij} = w_{ij} + r_i - r_j$

Suppose path $p_1(a \rightarrow b)$ is better than path $p_2(a \rightarrow b)$

e.g.: $w_{14} + w_{43} + w_{36} < w_{15} + w_{56}$

Compare paths with new weights:

$(w_{14} + r_1 - r_4) + (w_{43} + r_4 - r_3) + (w_{36} + r_3 - r_6)$ vs $(w_{15} + r_1 - r_5) + (w_{56} + r_5 - r_6)$
For every vertex \( k (1 \leq k \leq V) \)

let \( r_k \) be some real number.

Define \[ w'_{ij} = w_{ij} + r_i - r_j \]

Suppose path \( p_1(a \to b) \) is better than path \( p_2(a \to b) \)

\[ \text{e.g.} \quad w_{14} + w_{43} + w_{36} < w_{15} + w_{56} \]

Compare paths with new weights:

\[
(w_{14} + r_1 - r_4) + (w_{43} + r_4 - r_3) + (w_{36} + r_3 - r_6) \quad \text{vs} \quad (w_{15} + r_1 - r_5) + (w_{56} + r_5 - r_6)
\]
For every vertex \( k \) \((1 \leq k \leq V)\), let \( r_k \) be some real number.

Define \[ w'_{ij} = w_{ij} + r_i - r_j \]

Suppose path \( p_1(a \rightarrow b) \) is better than path \( p_2(a \rightarrow b) \)

e.g.: \( w_{14} + w_{43} + w_{36} < w_{15} + w_{56} \)

Compare paths with new weights:

\[
(w_{14} + r_1 - r_4) + (w_{43} + r_4 - r_3) + (w_{36} + r_3 - r_6) \quad \text{vs} \quad (w_{15} + r_1 - r_5) + (w_{56} + r_6 - r_5)
\]
For every vertex $k$ (1 ≤ $k$ ≤ $V$)
let $r_k$ be some real number.

Define $w'_{ij} = w_{ij} + r_i - r_j$

Suppose path $p_1(a \to b)$ is better
than path $p_2(a \to b)$

\[ w_{14} + w_{43} + w_{36} < w_{15} + w_{56} \]

Compare paths with new weights: $p_1$ still better than $p_2$

\[ (w_{14} + (r_1 - r_4)) + (w_{43} + (r_4 - r_3)) + (w_{36} + (r_3 - r_6)) < (w_{15} + (r_1 - r_5)) + (w_{56} + (r_5 - r_6)) \]
For every vertex \( k \) (\( 1 \leq k \leq v \)) let \( r_k \) be some real number.

Define \( w'_{ij} = w_{ij} + r_i - r_j \)

Also, \( \exists \) negative cycle in \( G' \) iff \( \exists \) negative cycle in \( G \)

\[
\sum_{\text{cycle}} w' = \sum_{\text{cycle}} w + \sum_{\text{cycle}} r_i - \sum_{\text{cycle}} r_j
\]

Conclusion: shortest path structure preserved
JOHNSON'S ALGORITHM

1 Bellman-Ford + ADJUST WEIGHTS + V · Dijkstra

- need weights \( \geq 0 \)

✓ need same shortest path structure
All $\omega_{sv} = 0$

(but the weights don't really matter)
All $\omega_{sv} = 0$

Define $r_v = \text{total weight of shortest path } s \rightarrow v$

Use Bellman-Ford
All $w_{sv} = 0$

same shortest path structure

Define $r_v =$ total weight of shortest path $s \rightarrow v$

Use Bellman-Ford
Assuming no negative cycles detected, we know B-F ends when no edge-relaxation causes a change.
Assuming no negative cycles detected,

we know B-F ends when no edge-relaxation causes a change.

\[ r_a + w_{ab} \geq r_b \]
All $w_{sv} = 0$

\[ \text{same shortest path structure} \]

Define \( r_v = \text{total weight of shortest path } s \rightarrow v \)

Use Bellman-Ford

Assuming no negative cycles detected,

we know B-F ends when no edge-relaxation causes a change.

\[ \forall \text{ edge } ab, \quad r_a + w_{ab} \geq r_b \]

By definition \( w'_{ab} = w_{ab} + r_a - r_b \)
All $\omega_{sv} = 0$

\[ \rightarrow \text{ same shortest path structure} \]

Define $r_v = \text{total weight of shortest path } s \rightarrow v$

Use Bellman-Ford

Assuming no negative cycles detected,

we know B-F ends when no edge-relaxation causes a change.

\[ \forall \text{ for every edge } ab, \quad r_a + \omega_{ab} \geq r_b \]

\[ \iff \omega_{ab} \geq r_b - r_a \]

By definition $\omega'_{ab} = \omega_{ab} + r_a - r_b$
All $\omega_{sv} = 0$

same shortest path structure

Define $r_v = \text{shortest path } s \rightarrow v$

Use Bellman-Ford

Assuming no negative cycles detected,

we know B-F ends when no edge-relaxation causes a change.

\[
\text{for every edge } ab, \quad r_a + \omega_{ab} \geq r_b
\]

\[
\omega_{ab} \geq r_b - r_a
\]

By definition $\omega'_{ab} = \omega_{ab} + r_a - r_b \geq (r_b - r_a) + r_a - r_b$
All $w_{sv} = 0$ 

same shortest path structure

Define $r_v$ = total weight of shortest path $s \rightarrow v$

Use Bellman-Ford

Assuming no negative cycles detected,

we know B-F ends when no edge-relaxation causes a change.

\[ \text{for every edge } ab, \quad r_a + w_{ab} \geq r_b \]

\[ w_{ab} \geq r_b - r_a \]

By definition $w'_{ab} = w_{ab} + r_a - r_b \geq (r_b - r_a) + r_a - r_b = 0$
JOHNSON'S ALGORITHM

= 1 Bellman-Ford + ADJUST WEIGHTS + V·Dijkstra

✓ need weights $\geq 0$

✓ need same shortest path structure
JOHNSON'S ALGORITHM

\[ \approx 1 \text{ Bellman-Ford} + \text{ADJUST WEIGHTS} + V \cdot \text{Dijkstra} \]

- need weights \( \geq 0 \)
- need same shortest path structure

Time: \( O(V \cdot E) + O(E) + V \cdot \text{Dijkstra} \)
JOHNSON'S ALGORITHM

& 1 Bellman-Ford + ADJUST WEIGHTS + V · Dijkstra

✓ need weights ≥ 0
✓ need same shortest path structure

Time: \( O(V \cdot E) + O(E) + V \cdot Dijkstra \)

\( V \cdot O(E \log V) \) or \( V \cdot O(E + V \log V) \)
JOHNSON'S ALGORITHM

\[ 1 \text{ Bellman-Ford} + \text{ADJUST WEIGHTS} + V \cdot \text{Dijkstra} \]

- need weights \( \geq 0 \)
- need same shortest path structure

Time: \( O(V \cdot E) + O(E) + V \cdot O(E \log V) \) or \( V \cdot O(E + V \log V) \)