Midterm 2 — next Monday

Conflict Tue AM

Review Thu + Fri

<u>Logistical changes</u>

+ PDF only
+ stop exam at 9:00
+ scan exam <u>only</u>

All-pairs shortest paths

Input: $G=(V,E)$ dir. graph
$w(e)$ weight for each edge e.

Output: $dist[1..V, 1..V]$ — shortest path lengths
$pred[1..V, 1..V]$ — predecessor

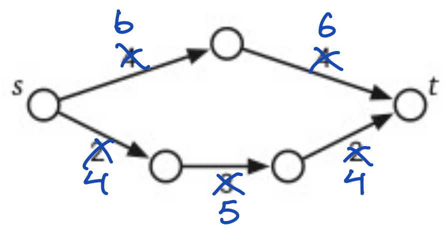
```
ObviousAPSP(V, E, w):
  for every vertex s
      dist[s, ·] ← SSSP(V, E, w, s)
```

$E = \Theta(V^2)$
<u>Dense</u>

	unweighted	— BFS	— $O(VE)$	$O(V^3)$
	dag	— DFS/DP	— $O(VE)$	$O(V^3)$
Sparse $\Longrightarrow$	non-neg wts	— Dijkstra	— $O(VE \log V)$	$O(V^3 \log V)$
	arb. wts	— BellmanFord	— $O(V^2E)$	$O(V^4)$

<u>Chan et al.</u> $O(V^3/\log^2 V)$ time

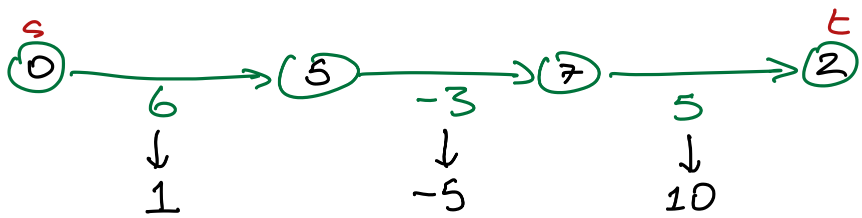
Fastest?? $O(V^{2.9999})$??



Repricing/Reweighting

"Price"

$$w'(u\to v) = \pi(u) + w(u\to v) - \pi(v)$$



$$w'(s\leadsto t) = \pi(s) + w(s\leadsto t) - \pi(t)$$

Compute dist(s,v) for every node v $\quad O(VE)$

set $\pi(v) \leftarrow \mathrm{dist}(s,v)$

tense: $\mathrm{dist}(s,v) > \mathrm{dist}(s,u) + w(u\to v)$

$w'(u\to v) < 0$

```
JOHNSONAPSP(V, E, w):
  ⟨⟨Add an artificial source⟩⟩
  add a new vertex s
  for every vertex v
      add a new edge s→v
      w(s→v) ← 0
  ⟨⟨Compute vertex prices⟩⟩
  dist[s,·] ← BELLMANFORD(V, E, w, s)
  if BELLMANFORD found a negative cycle
      fail gracefully
  ⟨⟨Reweight the edges⟩⟩
  for every edge u→v ∈ E
      w′(u→v) ← dist[s,u] + w(u→v) − dist[s,v]
  ⟨⟨Compute reweighted shortest path distances⟩⟩
  for every vertex u
      dist′[u,·] ← DIJKSTRA(V, E, w′, u)
  ⟨⟨Compute original shortest-path distances⟩⟩
  for every vertex u
      for every vertex v
          dist[u,v] ← dist′[u,v] − dist[s,u] + dist[s,v]
```

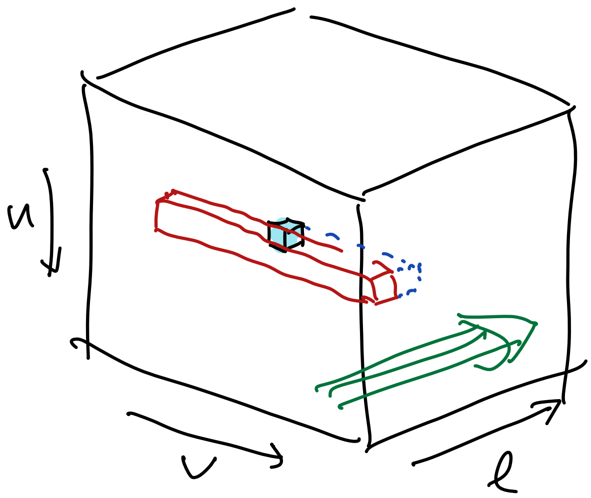
$O(VE \log V)$

$$dist(u,v) = \begin{cases} 0 & \text{if } u = v \\ \min\limits_{x \to v} \big(dist(u,x) + w(x \to v)\big) & \text{otherwise} \end{cases}$$

∞ loops!

dist(u,v,ℓ) = length of shortest path from u to v with ≤ ℓ edges.

$$dist(u,v,\ell) = \begin{cases} 0 & \text{if } \ell = 0 \text{ and } u = v \\ \infty & \text{if } \ell = 0 \text{ and } u \neq v \\ \min \left\{ \begin{array}{c} dist(u,v,\ell-1) \\ \min\limits_{x \to v} \big(dist(u,x,\ell-1) + w(x \to v)\big) \end{array} \right\} & \text{otherwise} \end{cases}$$

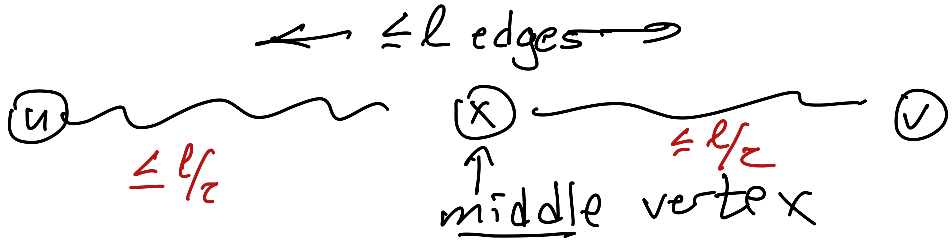


for ℓ ← 0 to V−1
  for all verts u
    for all verts v
      recurrence

$O(V^4)$ if dense

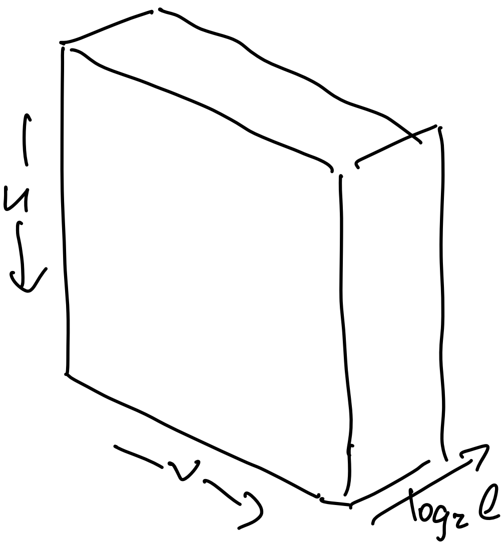
$V^2 \sum_v \deg(v) = O(V^2 E)$ if sparse

$\boxed{\text{Bellman-Ford} \times V}$



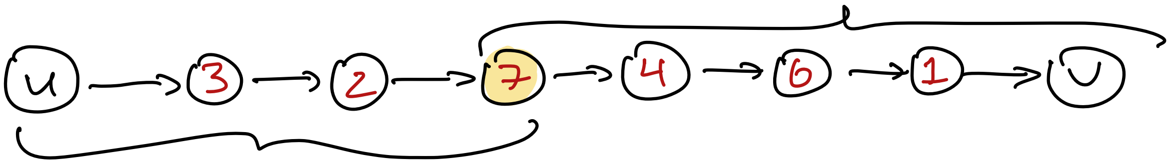
$$dist(u, v, \ell) = \begin{cases} w(u \to v) & \text{if } i = 1 \\ \min\limits_{x} \big(dist(u, x, \ell/2) + dist(x, v, \ell/2) \big) & \text{otherwise} \end{cases}$$

$V/2^i$ $\log_2 V$ different values



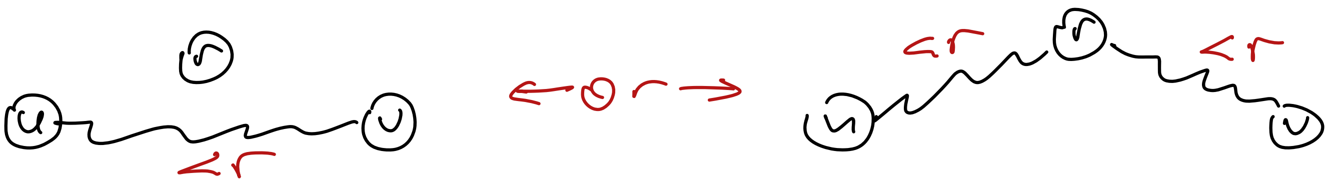
$O(V^3 \log V)$ time

```
LEYZOREKAPSP(V, E, w):
  for all vertices u
      for all vertices v
          dist[u, v] ← w(u→v)
  for i ← 1 to ⌈lg V⌉                  ⟨⟨ℓ = 2^i⟩⟩
      for all vertices u
          for all vertices v
              for all vertices x
                  if dist[u, v] > dist[u, x] + dist[x, v]
                      dist[u, v] ← dist[u, x] + dist[x, v]
```

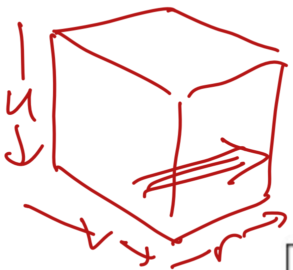


$dist(u,v,r)$ = length of the shortest path from u to v where all interior vertices have index $\le r$

We want $dist(u,v,V)$ for all u and v.



$$dist(u,v,r) = \begin{cases} w(u\to v) & \text{if } r = 0 \\ \min\left\{ \begin{array}{c} dist(u,v,r-1) \\ dist(u,r,r-1) + dist(r,v,r-1) \end{array} \right\} & \text{otherwise} \end{cases}$$




```
FLOYDWARSHALL(V, E, w):
  for all vertices u
      for all vertices v
          dist[u, v] ← w(u→v)
  for all vertices r  ← 1 to V
      for all vertices u
          for all vertices v
              if dist[u, v] > dist[u, r] + dist[r, v]
                  dist[u, v] ← dist[u, r] + dist[r, v]
```

$O(V^3)$

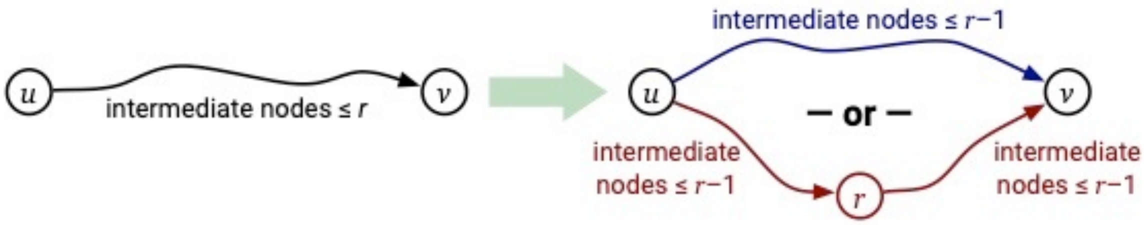
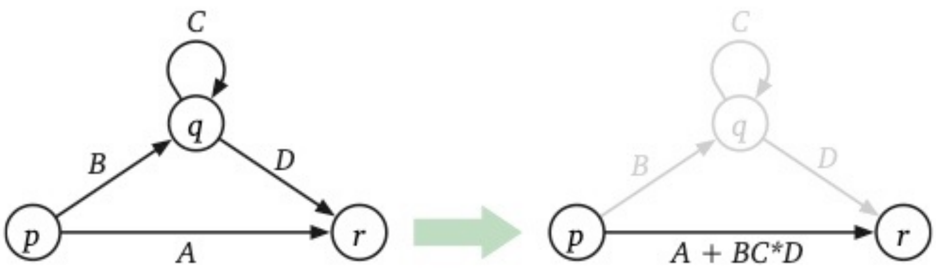


Figure 9.3. Recursive structure of the restricted shortest path $\pi(u, v, r)$.



Kleene's algorithm NFA $\longrightarrow$ reg. expressions

uses same pattern as Floyd-Warshall

$O(4^n)$ time