Midterm 2   Monday Nov. 6   7-9pm

recursion — D&C, backtracking, DP

graphs — traversal/connectivity, top. sort, strong comps, shortest paths

Thursday + Friday — review sessions

Thu PM, Sat PM, Mon AM — "study parties"

Sat 5:30 — HKN review session

2 practice exams   ∞ practice problems

Locations on web site

Conflict   DRES

Single-Source Shortest Paths

Input: Graph $G=(V,E)$ directed
$w: E \to \mathbb{R}$
source vertex $s \in V$

Output: For every vertex $v \in V$
$v.dist$ = length of shortest path from s to v
$v.pred$ = 2nd to last vertex on that path

Unweighted: BFS $O(V+E) = O(E)$

DAG: DFS/topsort $O(V+E)$

Non-neg weights: Dikstra $O(E \log V)$ or $O(E + V\log V)$

Arbitrary: Bellman-Ford $O(VE)$

```
BELLMANFORD(s)
  INITSSSP(s)
  repeat V − 1 times
      for every edge u→v
          if u→v is tense
              RELAX(u→v)
  for every edge u→v
      if u→v is tense
          return "Negative cycle!"
```

All-pairs shortest paths

Output: $dist[1..V, 1..V]$

$dist[u,v]$ = length of shortest $u \leadsto v$ path

$O(V^3)$ time   Floyd-Warshall

After i iterations of main loop:

$v.dist \le$ length of shortest path from s to v with $\le i$ edges.

Bellman-Ford Again

Given G, w, s

Build a dag $G' = (V', E')$

- $V' = V \times \{0, 1, \dots, V-1\}$ ← $|V'| = O(V^2)$
- $E' = \{(u,i) \to (v, i+1) \mid u \to v \in E \text{ and } 0 \le i < V-1\}$ ↖ $|E'| = O(VE)$
- $w'((u,i) \to (v,i+1)) = w(u \to v)$

This is a dag because 2nd component increases

Walk in G: $s \to v_1 \to v_2 \to \cdots \to v_\ell$

$\updownarrow$ same total weight

Path in G': $(s,0) \to (v_1, 1) \to (v_2, 2) \to \cdots \to (v_\ell, \ell)$

Shortest path from s to v in G

$\updownarrow$

Shortest path from $(s,0)$ to some (v,ℓ) in G'

We can compute $dist_G(s,v)$

by running DAG-SSSP algo from $(s,0)$ in G'

$O(V' + E')$ time

$= O(V^2 + VE)$

$= O(VE)$ if G is connected.

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

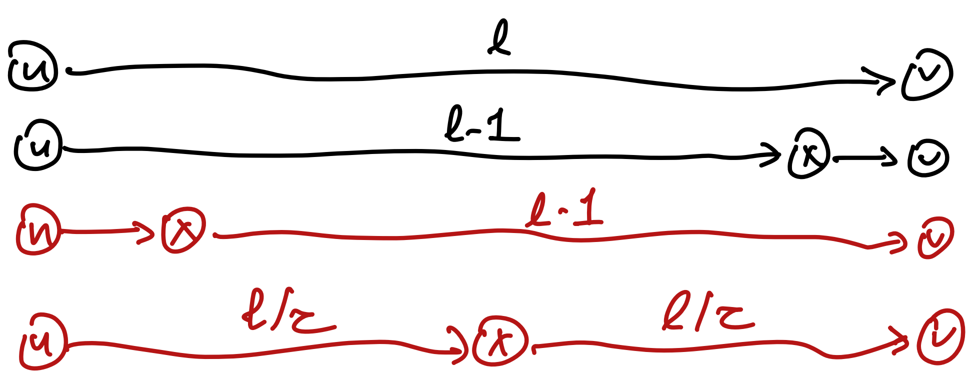
→ Unweighted: $O(VE)$ BFS
→ DAG $O(VE)$ DFS
→ non-neg wts: $O(VE \log V)$ Dijkstra
general: $O(V^2E)$ BF

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

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

```
AllPairsBellmanFord(V, E, w):
    for all vertices u
        for all vertices v
            if u = v
                dist[u, v] ← 0
            else
                dist[u, v] ← ∞
    for ℓ ← 1 to V − 1
        for all vertices u
            for all edges x→v
                if dist[u, v] > dist[u, x] + w(x→v)
                    dist[u, v] ← dist[u, x] + w(x→v)
```

$O(V^2E)$



$$dist(u,v,\ell) = \begin{cases} 0 & \text{if } u=v \\ w(u\to v) & \text{if } \ell=1 \\ \min_x \big(dist(u,x,\ell/2)+dist(x,v,\ell/2)\big) & \text{otherwise} \end{cases}$$

count variables $\Rightarrow O(V^4)$ time

but ℓ has only $O(\log V)$ values!

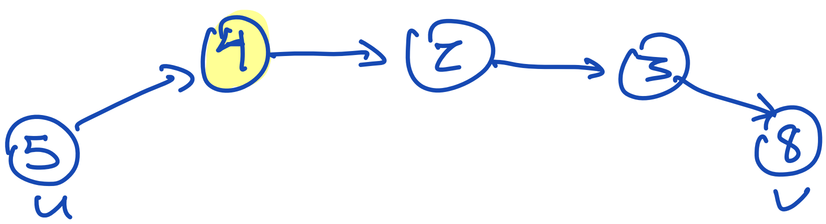
$\Rightarrow O(V^3 \log V)$

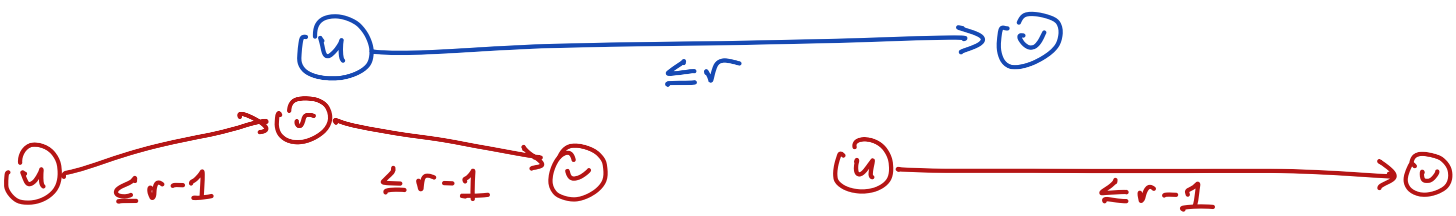
```
LEYZOREKAPSP(V, E, w):
  for all vertices u
      for all vertices v
          dist[u, v] ← w(u→v)
  for i ← 1 to ⌈lg V⌉              ⟨⟨ℓ = 2^i⟩⟩
      for all vertices u
          for all vertices v
              for all vertices x
                  if dist[u, v] > dist[u, x] + dist[x, v]
     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

$\log V$

$O(V^3 \log V)$

$dist(u,v,r)$ = length of shortest path from u to v through vertices $\leq r$





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

$\pi(u,v,r)$ is the shortest path (if any) from u to v that passes through only vertices numbered at most r.

```
FloydWarshall(V, E, w):
  for all vertices u
    for all vertices v
      dist[u, v] ← w(u→v)
  for all vertices r
    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]
```

$\boxed{O(V^3) \text{ time}}$

```
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]
```

```
FloydWarshall(V, E, w):
  for all vertices u
      for all vertices v
          dist[u, v] ← w(u→v)
  for all vertices r
      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]
```