HW 9 due today Nov 3.

Midterm 2 Mon Nov 9 7pm - 9pm

C. Midterm 2 Tue Nov 10

Practice 1) Thur Nov 5

2) Fri Nov 6

3) Sun Nov 8 (Siebel 0216)

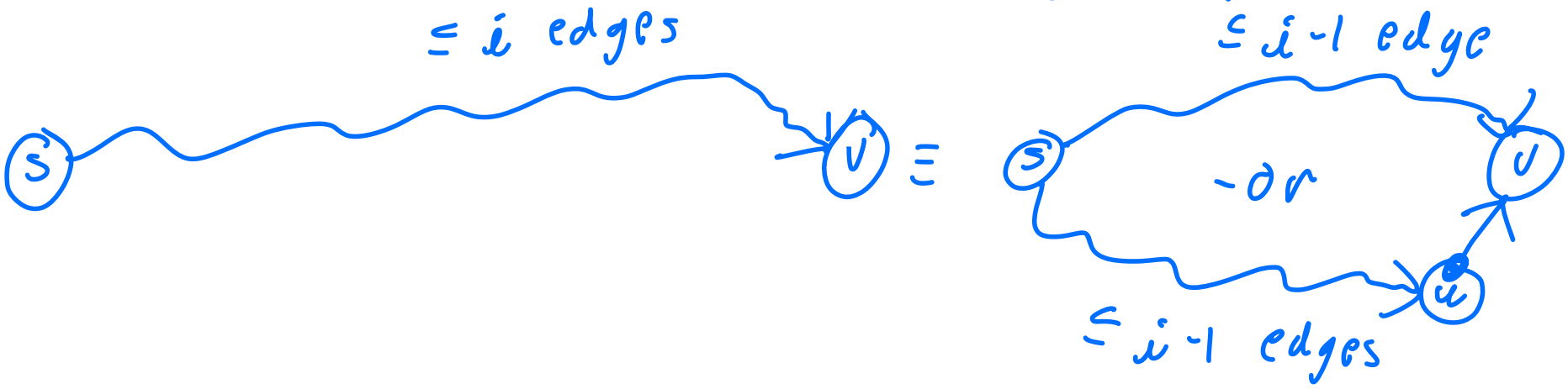
Given directed graph $G = (V, E)$ with real edge weights $w : E \to \mathbb{R}$ & a vertex $s \in V$.

distance(v) : distance from s to v

$$distance(v) = \begin{cases} 0 & \text{if } v = s \\ \min_{u \to v} (distance(u) + w(u \to v)) & \text{o.w.} \end{cases}$$

$distance_{\le i}(v)$: min length of any (s,v)-walk with <u>at most</u> i edges.

$v.dist = distance_{\le |V|-1}(v)$ (assuming no negative weight cycles)



$$distance_{\le i}(v) = \begin{cases} 0 & \text{if } i = 0 \text{ and } v = s \\ \infty & \text{if } i = 0 \text{ and } v \neq s \\ \min\left\{ distance_{\le i-1}(v), \min_{u \to v} \left(distance_{\le i-1}(u) + w(u \to v) \right) \right\} & \text{o.w.} \end{cases}$$

store in $distance[0..|V|-1, V]$

(i, v)

BellmanDP(s):

distance[0, s] $\leftarrow$ 0

```
    for each vertex v ≠ s
        distance[0, v] ← ∞
    for i ← 1 to |V| − 1
        for each vertex v
            distance[i, v] ← distance[i − 1, v]
            for each edge u → v
                if distance[i, v] > distance[i − 1, u] + w(u → v)
                    distance[i, v] ← distance[i − 1, u] + w(u → v)

    for each vertex v
        v.dist ← v.distance[|V| − 1, v]
```

Time: $O(|E||V|)$ (assuming G is connected)

BellmanDP2(s):

```
    distance[0, s] ← 0
    for each vertex v ≠ s
        distance[0, v] ← ∞
    for i ← 1 to |V| − 1
        for each vertex v
            distance[i, v] ← distance[i − 1, v]
        👉 for each edge u → v
            👉 if distance[i, v] > distance[i − 1, u] + w(u → v)
                👉 distance[i, v] ← distance[i − 1, u] + w(u → v)

    for each vertex v
        v.dist ← v.distance[|V| − 1, v]
```

BellmanDP3(s):

```
    distance[0, s] ← 0
    for each vertex v ≠ s
        distance[0, v] ← ∞
```

```
for i ← 1 to |V| − 1
    for each vertex v
        distance[i, v] ← distance[i − 1, v]
    for each edge u → v
        if distance[i, v] > distance[i, u] + w(u → v)
            distance[i, v] ← distance[i, u] + w(u → v)

for each vertex v
    v.dist ← v.distance[|V| − 1, v]
```

≤ length of shortest walks using ≤ i edges

```
BellmanDPFinal(s):
    distance[s] ← 0
    for each vertex v ≠ s
        distance[v] ← ∞
    for i ← 1 to |V| − 1
        for each edge u → v
            if distance[v] > distance[u] + w(u → v)
                distance[v] ← distance[u] + w(u → v)

    for each vertex v
        v.dist ← v.distance[v]
```

```
BellmanFord(s):
    InitSSSP(s)
    repeat |V| − 1 times
        for each edge u → v
```

```
        if u → v is tense
            Relax(u → v)

for each edge u → v
    if u → v is tense
        report "Negative cycle!"
```

All pairs shortest paths

Given directed $G = (V, E)$ with $w: E \to \mathbb{R}$.

Want to compute $dist(u, v)$: distance from u to v for all pairs $u, v \in V$.

Could solve SSSP for each choice of u.

Dijkstra

Positive weights: $O(|V| \cdot |E| \log |V|) \leq O(|V|^3 \log |V|)$

Negate weights (assume no negative cycles):

$$O(|V| \cdot |E||V|) \leq O(|V|^4)$$

$$dist(u,v) = \begin{cases} 0 & \text{if } v = u \\ \min_{x \to v} (dist(u,x) + w(x \to v)) & \text{o.w.} \end{cases}$$

$dist(u, v, l)$: length of shortest (u,v)-walk with $\leq l$ edges

$$dist(u,v,l) = \begin{cases} 0 & \text{if } l = 0 \text{ + } v = u \\ \infty & \text{if } l = 0 \quad v \neq u \\ \end{cases}$$

$$\left(\min \begin{cases} dist(u,v,\ell-1), \\ \min_{x \to v} \left(dist(u,x,\ell-1) + w(x \to v)\right) \end{cases}\right)$$

1954

ShimbelAPSP(G):

```
for each vertex u
    for each vertex v
        if u = v
            dist[u, v, 0] ← 0
        else
            dist[u, v, 0] ← ∞
for ℓ ← 1 to |V| − 1
    for each vertex u
        for each vertex v ≠ u
            dist[u, v, ℓ] ← dist[u, v, ℓ]
            for each edge x → v
                if dist[u, v, ℓ] > dist[u, x, ℓ − 1] + w(x → v)
                    dist[u, v, ℓ] ← dist[u, x, ℓ − 1] + w(x → v)
```

same cleanup as earlier

AllPairsBellmanFord(G):

```
for each vertex u
    for each vertex v
        if u = v
            dist[u, v] ← 0
        else
            dist[u, v] ← ∞
for ℓ ← 1 to |V| − 1
    for each vertex u
        for each edge x → v
            if dist[u, v] > dist[u, x] + w(x → v)
```

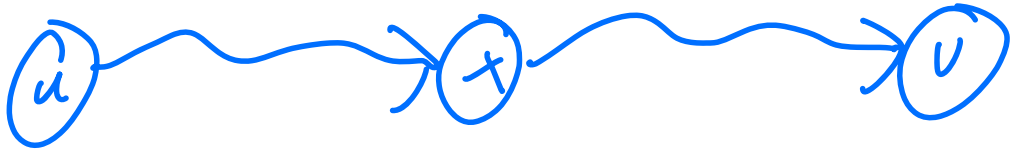
Time: $O(|V|^2|E|)$
$= O(|V|^4)$

$dist[u, v] \leftarrow dist[u, x] + w(x \rightarrow v)$

Just interleaved Bellman Fords!

(Assuming if $u \Rightarrow v \notin E$, then $w(u \Rightarrow v) = \infty$.)

$$dist(u, v, \ell) = \begin{cases} 0 & \text{if } \ell = 0 \text{ \& } v = u \\ \infty & \text{if } \ell = 0 \text{ \& } v \neq u \\ w(u \Rightarrow v) & \text{if } \ell = 1 \text{ \& } v \neq u \\ \min_x \left(dist(u, x, \ell/2) + dist(x, v, \ell/2) \right) \end{cases}$$



(Assume $|V| - 1$ is a power of 2)

Only need $O(\log_2 |V|)$ choices for ℓ to solve any $dist(u, v, |V|-1)$

```
FischerMeyerAPSP(V, E, w):
  for all vertices u
      for all vertices v
          dist[u, v, 0] ← w(u→v)
  for i ← 1 to ⌈lg V⌉                ⟨⟨ℓ = 2^i⟩⟩
      for all vertices u
          for all vertices v
              dist[u, v, i] ← ∞
              for all vertices x
                  if dist[u, v, i] > dist[u, x, i−1] + dist[x, v, i−1]
                      dist[u, v, i] ← dist[u, x, i−1] + dist[x, v, i−1]
```

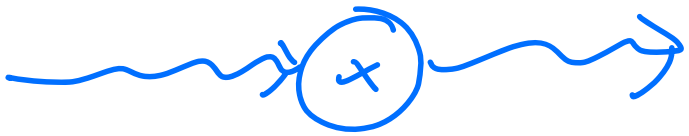
same cleanup

```
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]
```

Time: $O(|E||V| \log |V|)$

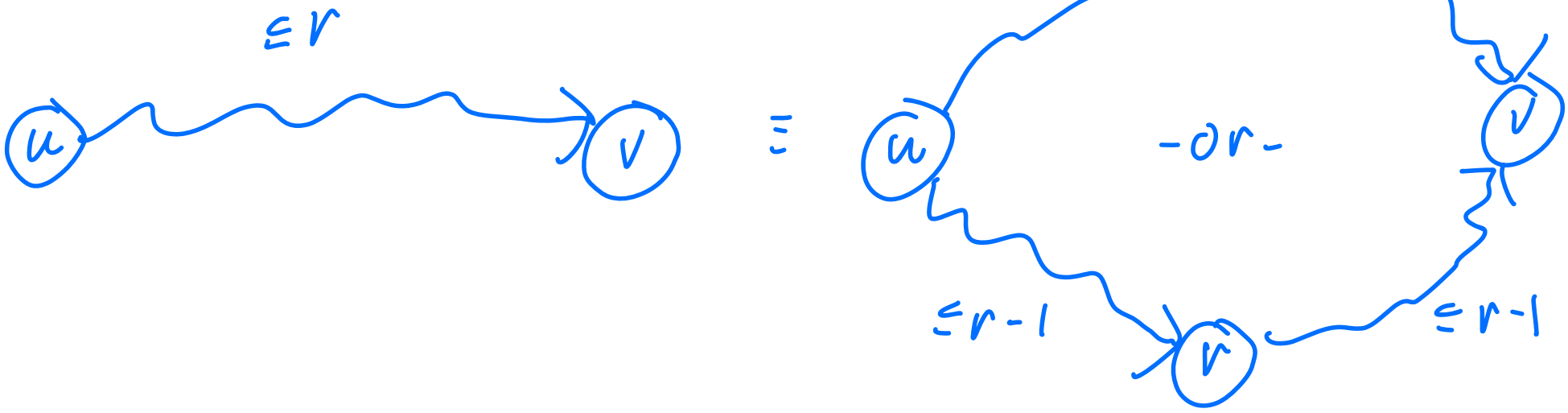
$= O(|V|^3 \log |V|)$

Say a path passes through vertex x if it enters & leaves x.



Number vertices 1 to $|V|$.

dist(u, v, r): length of shortest (u, v)-path passing through only vertice 1 through r.



$$dist(u,v,r) = \begin{cases} w(u \to v) & \text{if } r = 0 \\ \min\{ dist(u,v,r-1), \\ \quad dist(u,r,r-1) + dist(r,v,r-1) \} & \text{o.w.} \end{cases}$$

$$dist(u,v) = dist(u,v,|V|)$$

1951

```
KleeneAPSP(G):
    for each vertex u
        for each vertex v
            dist[u, v, 0] ← w(u → v)
    for r ← 1 to |V|
        for each vertex u          (vertex)
            for each ~~edge~~ v
                if dist[u, v, r − 1] < dist[u, r, r − 1] + dist[r, v, r − 1]
                    dist[u, v, r] ← dist[u, v, r − 1]
                else
                    dist[u, v, r] ← dist[u, r, r − 1] + dist[r, v, r − 1]
```

Time: $O(|V|^3)$

FloydWarshall(G):

```
for each vertex u
    for each vertex v
        dist[u, v] ← w(u → v)

for each vertex r
    for each vertex u
        for each ~~edge~~ v
            if dist[u, v] > dist[u, r] + dist[r, v]
                dist[u, v] ← dist[u, r] + dist[r, v]
```