Midterm 2 next next Mon Nov 11

This is the last MT2 lecture

Shortest Paths

Ford (1956): Every vertex maintains

dist(v) — upper bound on shortest path distance from s to v

pred(v) — predecessor of v on "shortest" path from s to v.

Init

pred(s) ← NULL
dist(s) ← 0
for all vertices v ≠ s
  dist(v) ← ∞
  pred(v) ← NULL

Edge u→v is tense iff

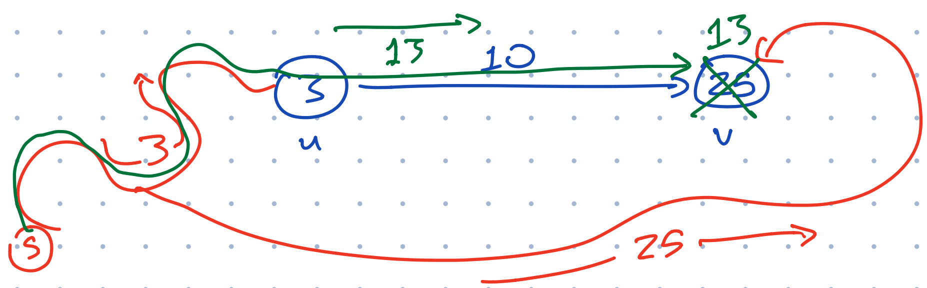
$dist(u) + w(u \to v) < dist(v)$

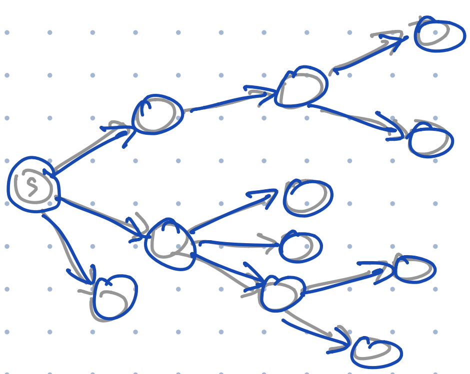
Relax(u→v):

$dist(v) \leftarrow dist(u) + w(u \to v)$
$pred(v) \leftarrow u$

GenericSP:

Init
while there is a tense edge
  relax any tense edge



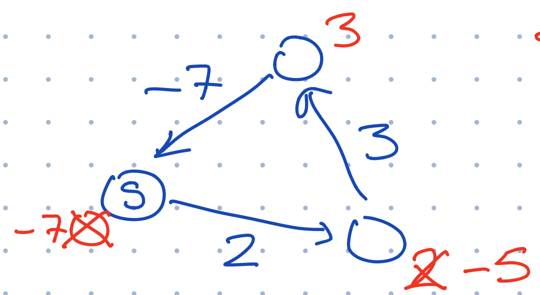


Unweighted — BFS $O(V+E)$

Dag, weighted — DFS/topsort $O(V+E)$

Weighted, no neg weights — Dijkstra $O(E \log V)$

Weighted — BellmanFord $O(EV)$

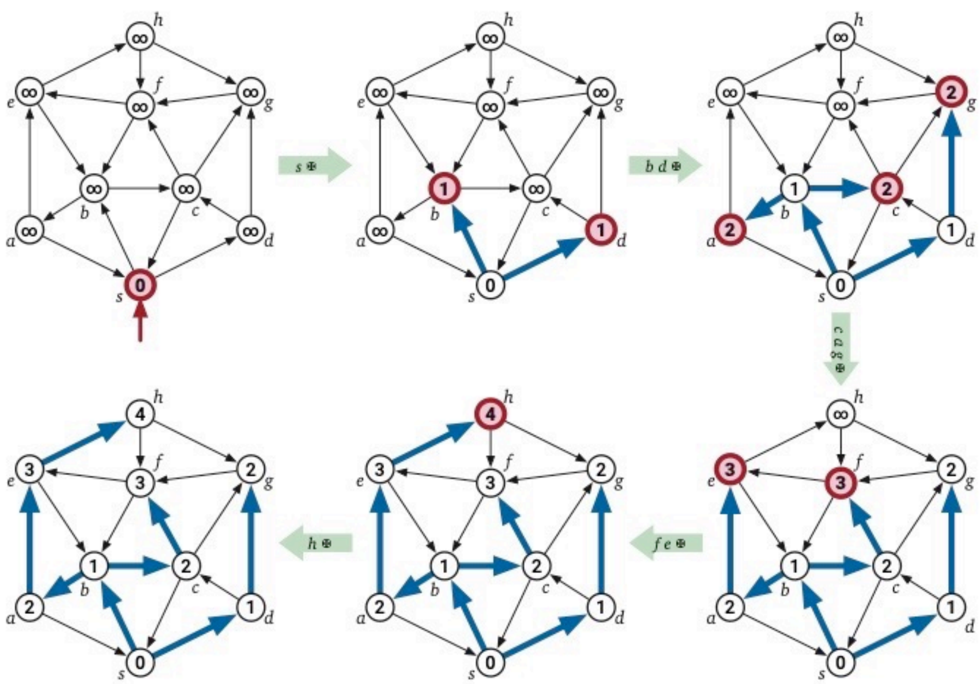


Basic SP algo assume no negative cycles

Easy modifications detect neg. cycles.

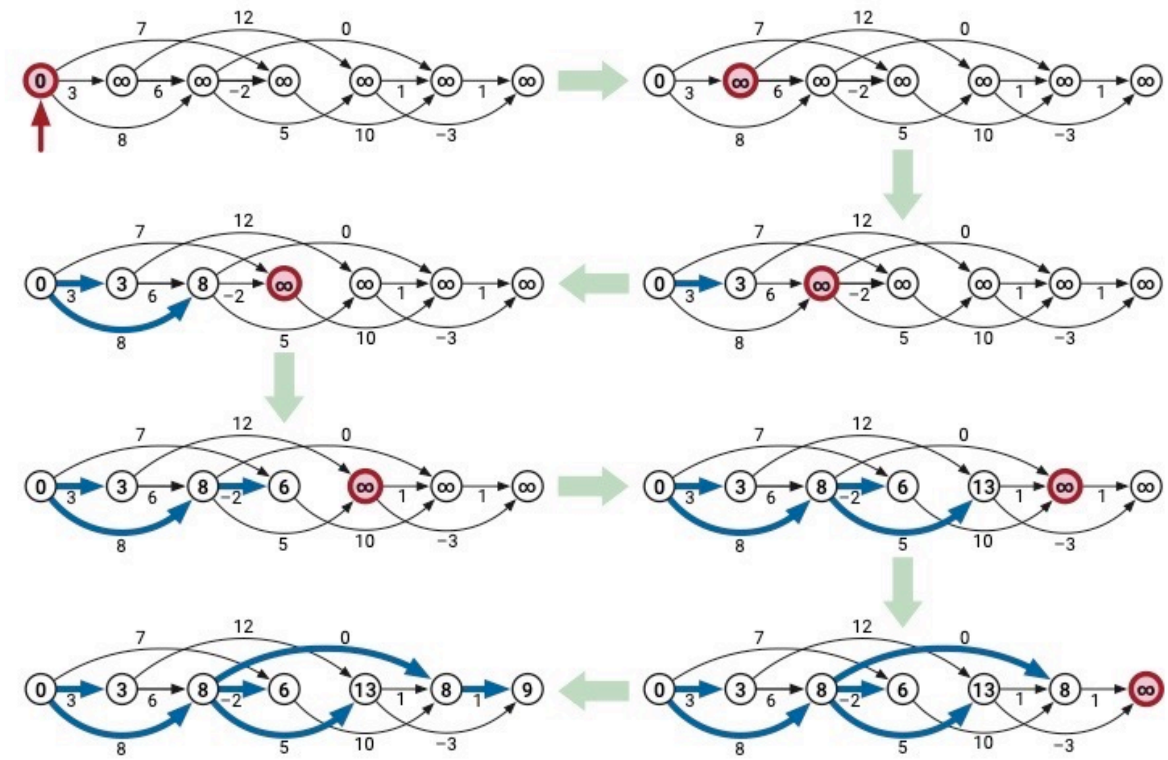
```
BFSWithToken(s):
  InitSSSP(s)
  Push(s)
  Push(✠)                       ⟨⟨start the first phase⟩⟩
  while the queue contains at least one vertex
      u ← Pull()
      if u = ✠
          Push(✠)               ⟨⟨start the next phase⟩⟩
      else
          for all edges u→v
              if dist(v) > dist(u) + 1          ⟨⟨if u→v is tense⟩⟩
                  dist(v) ← dist(u) + 1
                  pred(v) ← u                   ⟨⟨relax u→v⟩⟩
                  Push(v)
```

Every vertex is PULLED once
edge is RELAXED ≤ once | $O(V+E)$ time



```
DagSSSP(s):
  InitSSSP(s)
  for all vertices v in topological order
      for all edges u→v
          if u→v is tense
              Relax(u→v)
```

O(V+E) time



$$dist(v) = \begin{cases} 0 & \text{if } v = s \\ \min\limits_{u \to v} \left(dist(u) + w(u \to v) \right) & \text{otherwise} \end{cases}$$

shortest path distance from s to v

Always correct

Not always fast

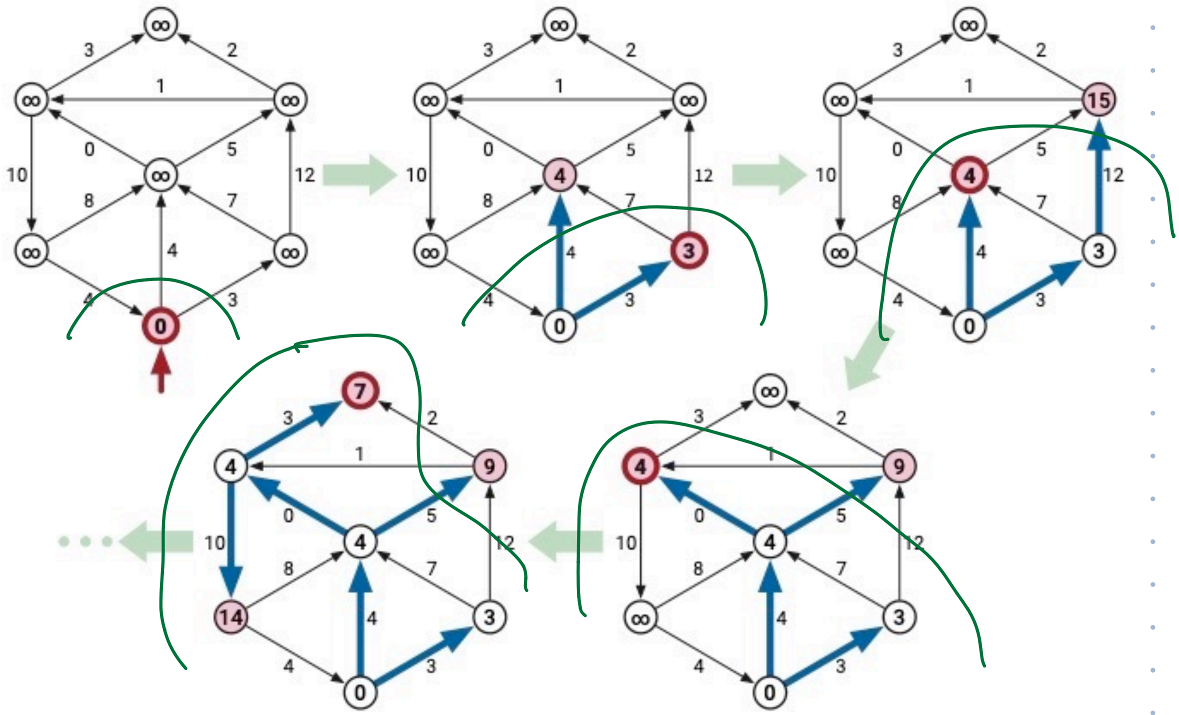
$\Theta(2^V)$ time

```
DIJKSTRA(s):
  INITSSSP(s)
  INSERT(s, 0)
  while the priority queue is not empty
      u ← EXTRACTMIN()
      for all edges u→v
          if u→v is tense
              RELAX(u→v)
              if v is in the priority queue
                  DECREASEKEY(v, dist(v))
              else
                  INSERT(v, dist(v))
```

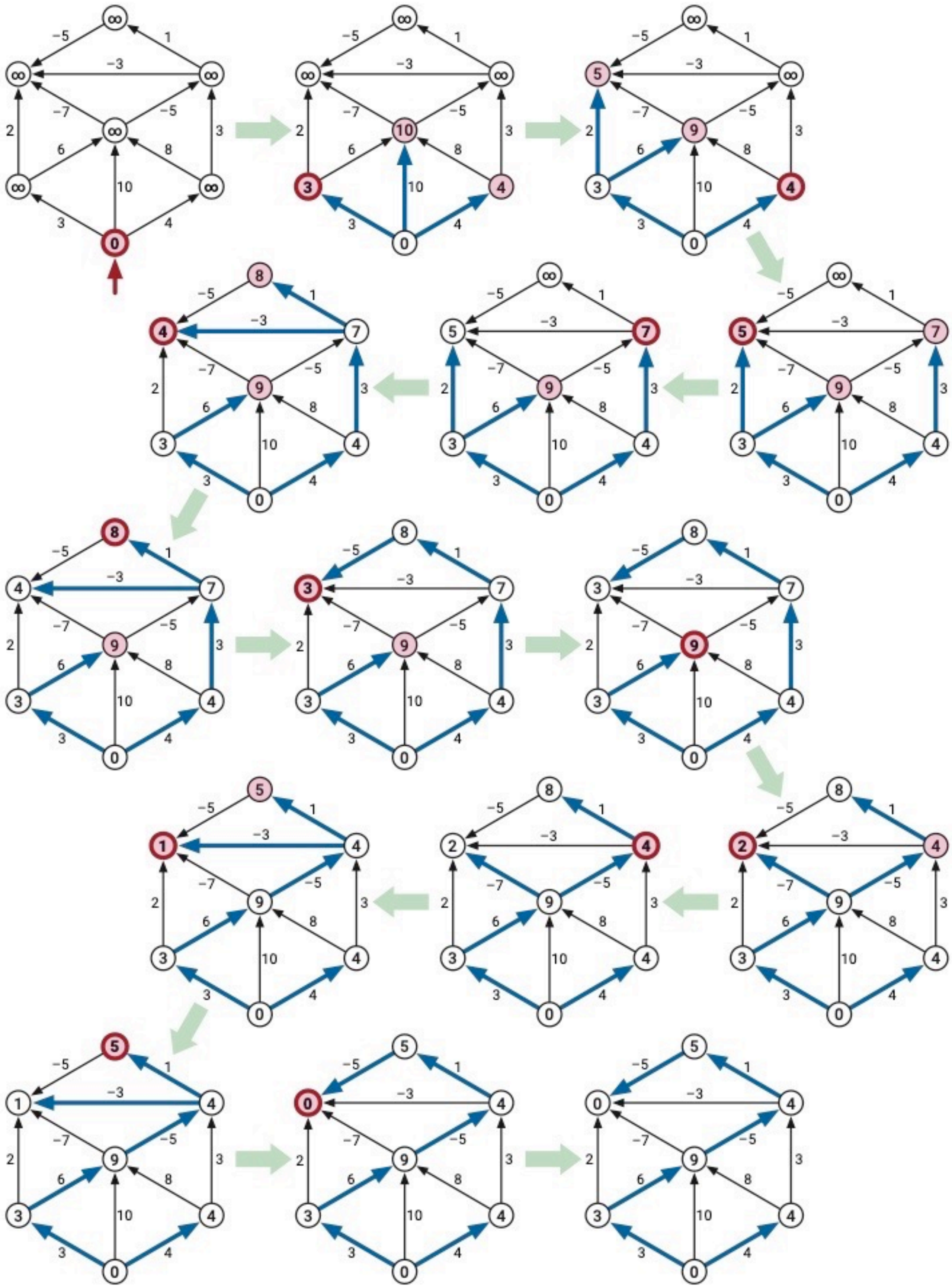
No neg edges:
- Every vertex is EXTRACTED once
- Every edge is relaxed ≤ once

$O(E \log V)$

PQ




```
NONNEGATIVEDIJKSTRA(s):
  INITSSSP(s)
  for all vertices v
      INSERT(v, dist(v))
  while the priority queue is not empty
      u ← EXTRACTMIN()
      for all edges u→v
          if u→v is tense
              RELAX(u→v)
              DECREASEKEY(v, dist(v))
```



```
BellmanFord(s)
  InitSSSP(s)
  while there is at least one tense edge
      for every edge u→v
          if u→v is tense
              Relax(u→v)
```

$O(VE)$ time

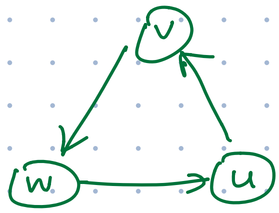
Let $\text{dist}_{\le i}(v)$ denote length of shortest path from s to v with $\le i$ edges.

<u>Lemma:</u> After i passes dist(v) and pred(v) are correct for all v s.t. shortest path $s \rightsquigarrow v$ has $\le i$ edges.

```
BellmanFord(s)
  InitSSSP(s)
  repeat V − 1 times            ← V
      for every edge u→v        ← E
          if u→v is tense
              Relax(u→v)
  for every edge u→v
      if u→v is tense
          return "Negative cycle!"
```

$O(VE)$ time

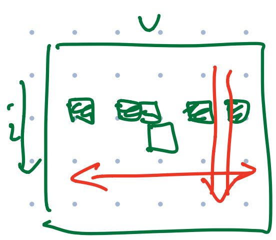
$$dist(v) = \begin{cases} 0 & \text{if } v = s \\ \min\limits_{u \to v} (dist(u) + w(u \to v)) & \text{otherwise} \end{cases}$$



Let $dist_{\le i}(v)$ denote length of shortest path from s to v with $\le i$ edges.

$$dist_{\le i}(v) = \begin{cases} 0 & \text{if } i = 0 \text{ and } v = s \\ \infty & \text{if } i = 0 \text{ and } v \ne s \\ \min \left\{ \begin{array}{c} dist_{\le i-1}(v) \\ \min\limits_{u \to v} (dist_{\le i-1}(u) + w(u \to v)) \end{array} \right\} & \text{otherwise} \end{cases}$$

We need $dist_{\le V-1}(u)$ for all vertices u.



BELLMANFORDDP(s)
  $dist[0, s] \leftarrow 0$
  for every vertex $v \ne s$
    $dist[0, v] \leftarrow \infty$
$V \rightarrow$ for $i \leftarrow 1$ to $V - 1$
    for every vertex v
      $dist[i, v] \leftarrow dist[i-1, v]$
$\le E$ times   for every edge $u \to v$
        if $dist[i, v] > dist[i-1, u] + w(u \to v)$
          $dist[i, v] \leftarrow dist[i-1, u] + w(u \to v)$

$O(VE)$ time

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

Floyd Warshall $O(V^3)$ time

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

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

```
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]
```

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

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