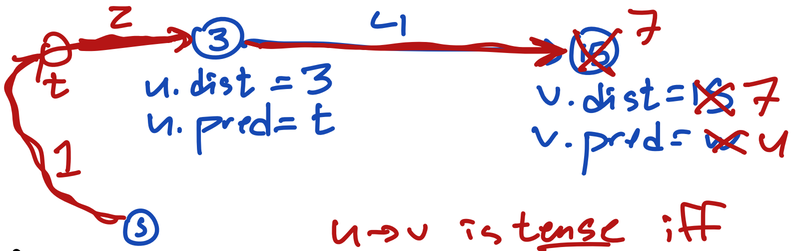
Shortest Paths

unweighted : BFS $O(V+E)$

dag : DFS $O(V+E)$

non-negative weights   Dijkstra $O(E \log V)$ ← assuming connected
$O((V+E) \log V)$

arbitrary   Bellman-Ford $O(VE)$



$u \to v$ is tense iff
$u.dist + w(u \to v) < v.dist$

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

DIJKSTRA(s):
  INITSSSP(s)
  INSERT(s, 0)
  while the priority queue is not empty
    $u \leftarrow$ EXTRACTMIN( )   ← ≤ V times
    for all edges $u \to v$
      if $u \to v$ is tense
        RELAX($u \to v$)   ← ≤ E times
        ~~if v is in the priority queue~~
          DECREASEKEY(v, $v.dist$)   ← ≤ E times
        ~~else~~
          ~~INSERT(v, $v.dist$)~~

"Best-first search"

INSERT(x, p)
EXTRACTMIN
DECREASEKEY(x, p)

Binary heap $O(\log V)$

priority(v) = $v.dist$

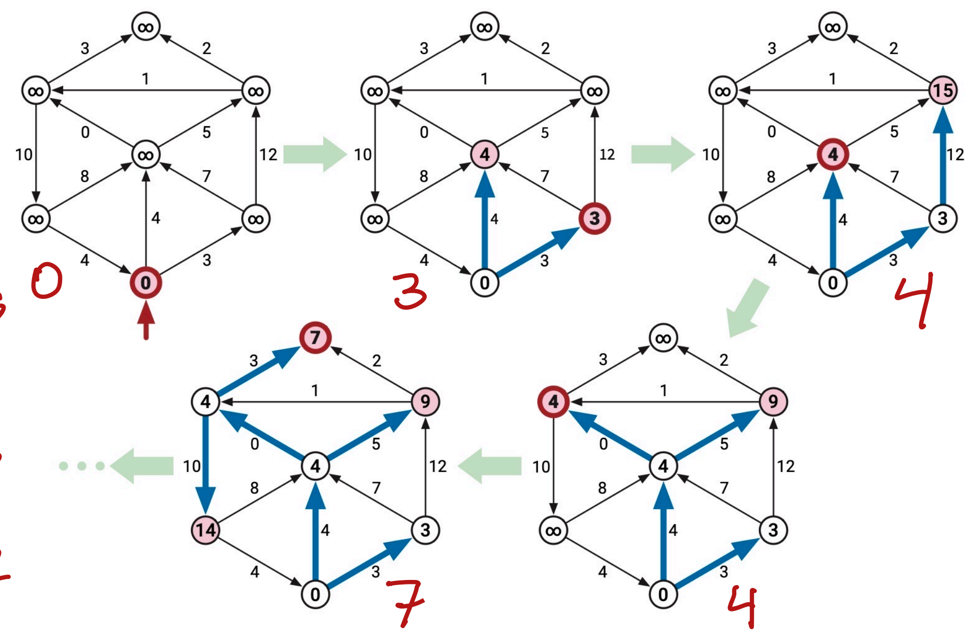
→ $O(V+E)$ PQ operations
⇒ $O(E \log V)$ time

PQ: min distance only increases

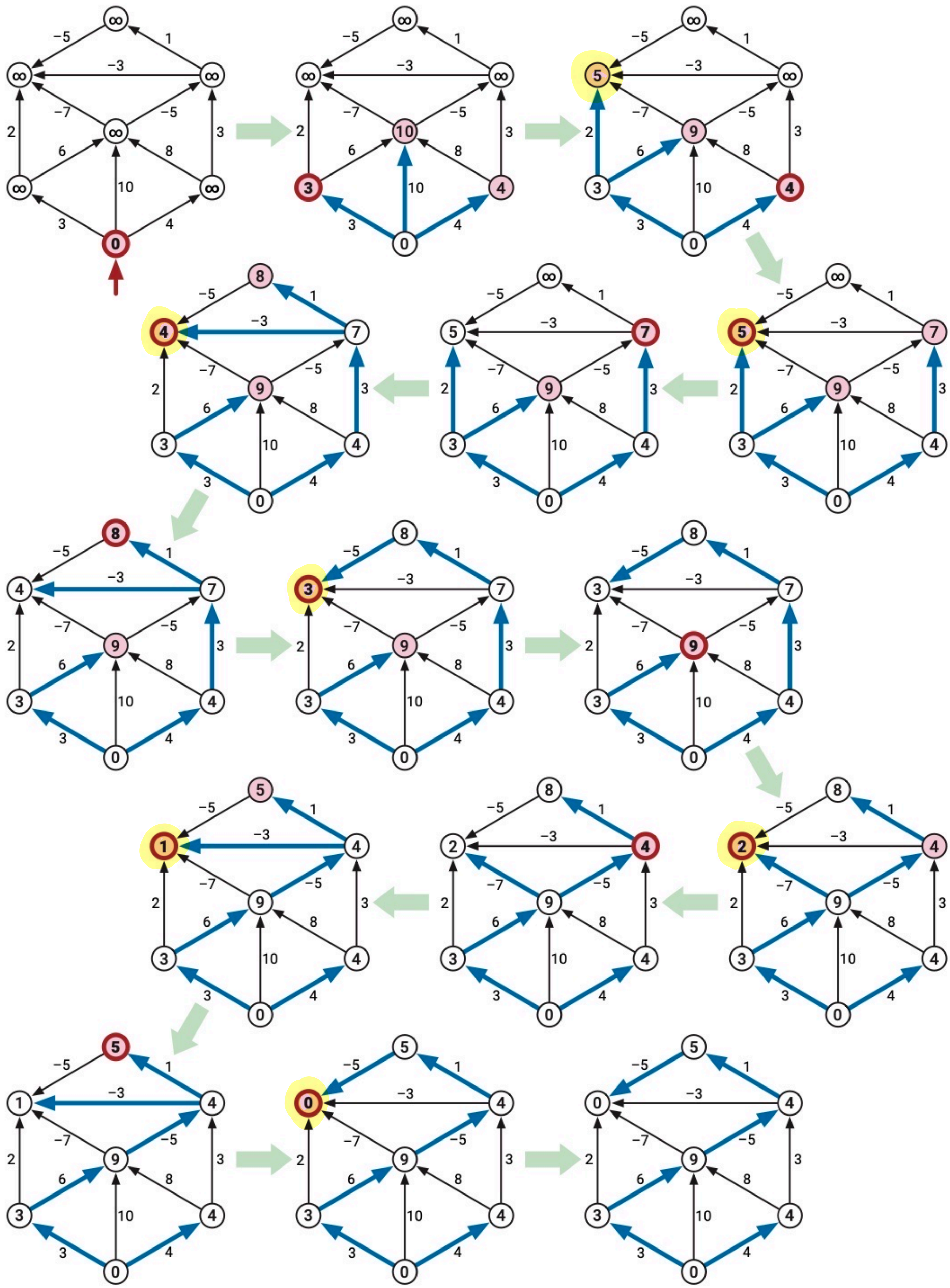
at each node: distance only decreases

⇓

Each vertex is EXTRACTED at most once



NONNEGATIVEDIJKSTRA(s):
  INITSSSP(s)
  for all vertices v
    INSERT(v, $v.dist$)
  while the priority queue is not empty
    $u \leftarrow$ EXTRACTMIN( )
    for all edges $u \to v$
      if $u \to v$ is tense
        RELAX($u \to v$)
        DECREASEKEY(v, $v.dist$)



Worst case Dijstra: $2^{\Theta(V)}$ $O(1)$ neg edges: $O(E \log V)$

BELLMAN-FORD: Relax *ALL* the tense edges, then recurse.

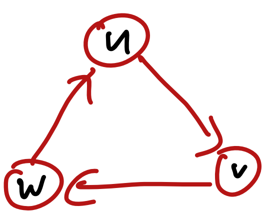
```
BELLMANFORD(s)
  INITSSSP(s)
  while there is at least one tense edge
      for every edge u→v
          if u→v is tense
              RELAX(u→v)
```

```
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!”
```

O(VE) time

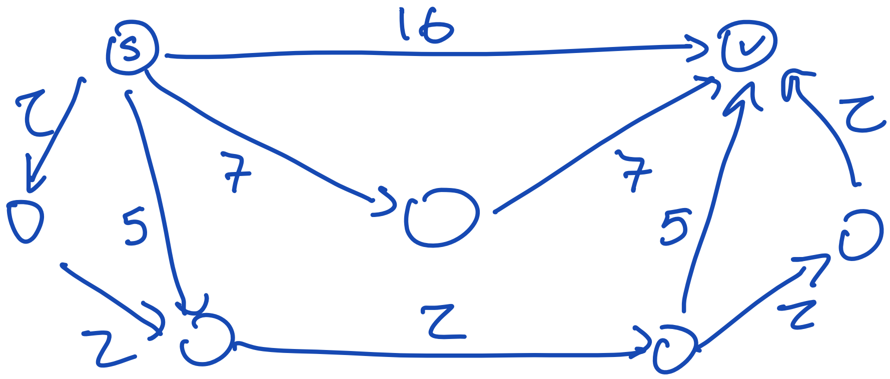
dist(v) = length of shortest path from s to v

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

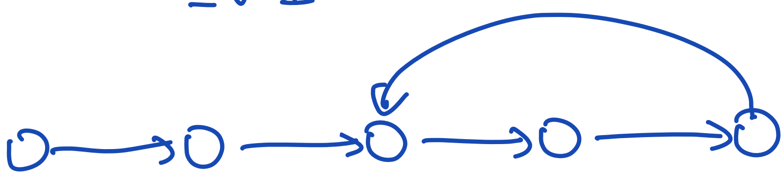


$dist_{\leq i}(v)$ = length of shortest path with at most i edges from s to v.

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



$dist(v) = dist_{\le V-1}(v)$ because sh paths have $\le V-1$ edges



v.dist[1..V] or dist[1..V, 1..V]

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

```
BellmanFordDP(s)
  dist[  s] ← 0
  for every vertex v ≠ s
      dist[  v] ← ∞
  for i ← 1 to V − 1


      for every edge u→v
          if dist[  v] > dist[      u] + w(u→v)
              dist[  v] ← dist[      u] + w(u→v)
```