| 6.006 | Lecture 14 | Nov. 1, 2016 |
|---|---|---|

**TODAY:** Shortest Paths via Bellman-Ford
→ algorithm
→ running time: $O(VE)$
→ correctness

- handling negative-weight cycles
- directed acyclic graphs: $O(V+E)$

**Recall:**

- $\delta(u,v) = \begin{cases} \inf\{w(p) \mid \text{path } p \text{ from } u \text{ to } v\} \\ \infty \text{ if no path from } u \text{ to } v \end{cases}$

  ($-\infty$ if neg.-weight cycle on the way) (path: can repeat vertices)

- SSSP: given edge-weighted directed graph $G=(V,E,w)$ & source $s \in V$, compute $\delta(s,v)$ for all $v \in V$, and shortest-path tree containing a shortest path from $s$ to each $v \in V$
- Relaxation Algorithm:

  initialization(V):
  for v in V:
      v.d = ∞
      v.parent = None
  s.d = ∅

  while some edge (u,v) has v.d > u.d + w(u,v):
      pick some edge (u,v)
      relax(u,v): if v.d > u.d + w(u,v):
          v.d = u.d + w(u,v)
          v.parent = u

- Safety Lemma: $v.d \geq \delta(s,v)$ for all $v \in V$ invariant throughout relaxation

**Bellman-Ford algorithm:** [Shimbel 1955; Ford 1956; Bellman 1958]

- relaxation algorithm for SSSP
- label edges: $E = \{e_1, e_2, \ldots, e_m\}$
- relaxation order:

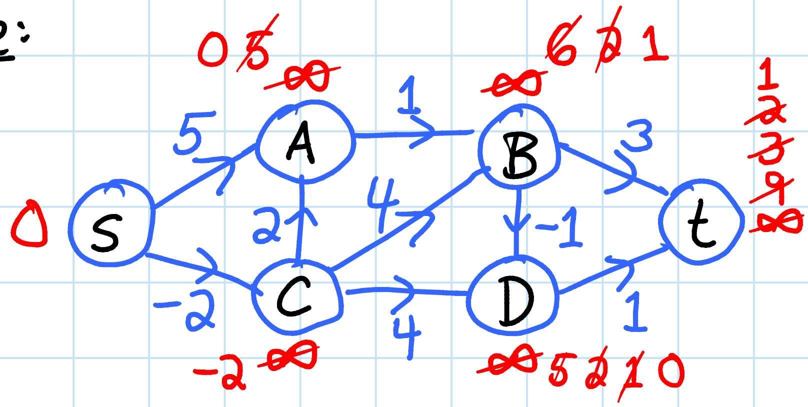
$$\left.\begin{array}{l} e_1, e_2, \ldots, e_m, \\ e_1, e_2, \ldots, e_m, \\ \vdots \\ e_1, e_2, \ldots, e_m \end{array}\right\} |V|-1 \text{ reps.}$$

**Algorithm:**

```
initialization(V)              - Θ(V)
for i from 1 to |V|-1:                      } Θ(VE)
    for (u,v) in E:            } Θ(E)
        relax(u,v)             - Θ(1)
handle negative-weight cycles (to be filled later)
```

Total: $\boxed{O(VE)}$

**Example:**



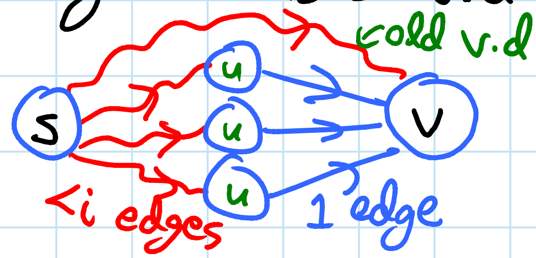

edges ordered top down, left → right

**In practice:** Internet's Routing Information Protocol & Interior Gateway Routing Protocol

- relaxation is local / distributed!

**Correctness Lemma:** after iteration $i$ of Bellman-Ford, $v.d \leq \min\{w(p) \mid \text{path } p \text{ from } s \text{ to } v \text{ using at most } i \text{ edges}\}$, for all $v \in V$

**Proof:** by induction on $i$
- **base case:** initialization covers paths of $\emptyset$ edges
- before iteration $i$: $v.d \leq \min\{w(p) \mid \cdots < i \text{ edges}\}$
- remains true: relaxation only decreases $v.d$
- iteration $i$ considers all paths with $\leq i$ edges:




□

**Corollary:** if $G$ has no negative-weight cycles then Bellman-Ford computes $v.d = \delta(s,v)$ for all $v \in V$

**Proof:**

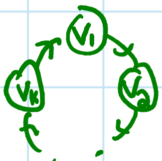
$\Rightarrow$ shortest paths are simple
↳ no repeated vertices

$\Rightarrow \leq |V|$ vertices, $< |V|$ edges
- Correctness Lemma $\Rightarrow v.d \leq \delta(s,v)$
- Safety Lemma $\Rightarrow v.d \geq \delta(s,v)$. □

# Handling negative-weight cycles:

**Version 1:** [textbook]

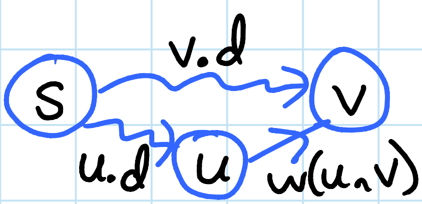
for $(u,v)$ in $E$: — $\Theta(E)$
    if $v.d > u.d + w(u,v)$: ← could relax
        report existence of neg.-weight cycle reachable from $s$




sum $\Delta$ ineq.

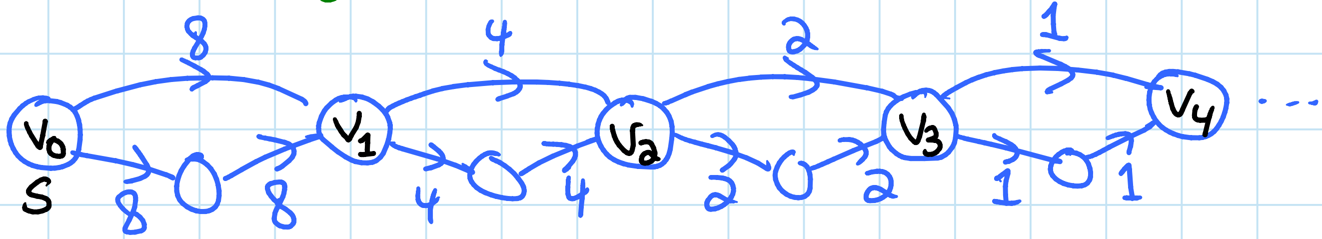
- if there's such a negative-weight cycle, one of its edges can always be relaxed once one of its $d$ values becomes finite
    $\Rightarrow$ find it
- else: $\underset{\delta(s,d)}{\overset{\shortparallel}{v.d}} \le \underset{\delta(s,u)}{\overset{\shortparallel}{u.d}} + w(u,v)$ by $\Delta$ inequality by corollary above

**Version 2:** for $j$ from 1 to $|V|$:
    for $(u,v)$ in $E$:
        if $v.d > u.d + w(u,v)$:
            $v.d = -\infty$
            $v.parent = u$
$\Bigg\}\ O(V \cdot E)$




- every negative-weight cycle reachable from $s$ has a relaxable edge $(u,v)$ (after B-F)
    $\Rightarrow$ set $v.d = -\infty$ in first round
- all nodes $x$ reachable from $v$ have $\delta(s,x) = -\infty$
    ↳ including $u$ (still safe)
- path from $v$ to $x$ simple $\Rightarrow$ $< |V|$ edges
    $\Rightarrow$ $|V|-1$ more rounds suffice
- parent pointers will form some cycles

**Example:** edges ordered left to right




| d: | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ |
| 0 | 8 | 8 | 12 | 12 | 14 | 14 | 15 | 15 |

**Why so fast?** DAG + topological sort order

**Shortest paths in a DAG:** one round of B-F

```
initialization(V)
topologically sort G            – O(V+E)
for u in V:  (in topo. sort order)  }
    for v in Adj[u]:    } deg(u)     } O(V+E)
        relax(u,v)                   }
```

– no (negative-weight) cycles to deal with

**Correctness:** $v.d = \delta(s,v)$ when $v$ visited by the outer for loop

**Proof:** by induction on position of $v$ in topo. order

- **base:** $s.d = 0$ (no cycles) & unreachable $v.d = \infty$
- when visiting $v$, already visited $u$ with $(u,v) \in E$
  - $\Rightarrow$ $u.d$ values correct by induction
  - $\Rightarrow$ $v.d = \min\{u.d + w(u,v) \mid (u,v) \in E\}$ correct: considers all paths $s \to v$. $\square$