Lecture 12: Bellman-Ford

Previously

- Weighted graphs, shortest-path weight, negative-weight cycles
- Finding shortest-path tree from shortest-path weights in $O(|V| + |E|)$ time
- DAG Relaxation: algorithm to solve SSSP on a weighted DAG in $O(|V| + |E|)$ time
- SSSP for graph with negative weights
  - Compute $\delta(s, v)$ for all $v \in V$ ($-\infty$ if v reachable via negative-weight cycle)
  - If a negative-weight cycle reachable from s, return one

Warmups

- **Exercise 1:** Given undirected graph G, return whether G contains a negative-weight cycle
- **Solution:** Return **Yes** if there is an edge with negative weight in G in $O(|E|)$ time :O
- So for this lecture, we restrict our discussion to **directed graphs**

- **Exercise 2:** Given SSSP algorithm A that runs in $O(|V|(|V| + |E|)$ time, show how to use it to solve SSSP in $O(|V||E|)$ time
- **Solution:** Run BFS or DFS to find the vertices reachable from s in $O(|E|)$ time
  - Mark each vertex v not reachable from s with $\delta(s, v) = \infty$ in $O(|V|)$ time
  - Make graph $G' = (V', E')$ with only vertices reachable from s in $O(|V| + |E|)$ time
  - Run A from s in G'.
  - G' is connected, so $|V'| = O(|E'|) = O(|E|)$ so A runs in $O(|V||E|)$ time
- Today, we will find a SSSP algorithm with this running time that works for general graphs!

Restrictions		SSSP Algorithm		
Graph	Weights	Name	Running Time $O(\cdot)$	Lecture
General	Unweighted	BFS	$\|V\| + \|E\|$	L09
DAG	Any	DAG Relaxation	$\|V\| + \|E\|$	L11
General	Any	Bellman-Ford	$\|V\| \cdot \|E\|$	L12 (Today!)
General	Non-negative	Dijkstra	$\|V\| \log \|V\| + \|E\|$	L13

Simple Shortest Paths

- If graph contains cycles and negative weights, might contain negative-weight cycles :(
- If graph does not contain negative-weight cycles, shortest paths are simple!
- **Claim 1:** If $\delta(s, v)$ is finite, there exists a shortest path to v that is **simple**
- **Proof:** By contradiction:
  - Suppose no simple shortest path; let π be a shortest path with fewest vertices
  - π not simple, so exists cycle C in π; C has non-negative weight (or else $\delta(s, v) = -\infty$)
  - Removing C from π forms path π' with fewer vertices and weight $w(\pi') \leq w(\pi)$ $\square$
- Since simple paths cannot repeat vertices, finite shortest paths contain at most $|V| - 1$ edges

Negative Cycle Witness

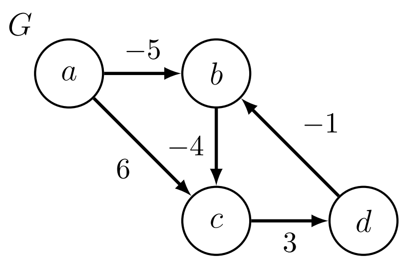
- ***k*-Edge Distance** $\delta_k(s, v)$: the minimum weight of any path from s to v using $\leq k$ edges
- **Idea!** Compute $\delta_{|V|-1}(s, v)$ and $\delta_{|V|}(s, v)$ for all $v \in V$
  - If $\delta(s, v) \neq -\infty$, $\delta(s, v) = \delta_{|V|-1}(s, v)$, since a shortest path is simple (or nonexistent)
  - If $\delta_{|V|}(s, v) < \delta_{|V|-1}(s, v)$
    * there exists a shorter non-simple path to v, so $\delta_{|V|}(s, v) = -\infty$
    * call v a (negative cycle) **witness**
  - However, there may be vertices with $-\infty$ shortest-path weight that **are not witnesses**
- **Claim 2:** If $\delta(s, v) = -\infty$, then v is reachable from a witness
- **Proof:** Suffices to prove: every negative-weight cycle reachable from s contains a witness
  - Consider a negative-weight cycle C reachable from s
  - For $v \in C$, let $v' \in C$ denote v's predecessor in C, where $\sum_{v \in C} w(v', v) < 0$
  - Then $\delta_{|V|}(s, v) \leq \delta_{|V|-1}(s, v') + w(v', v)$ (RHS weight of some path on $\leq |V|$ vertices)
  - So $\sum\limits_{v \in C} \delta_{|V|}(s, v) \leq \sum\limits_{v \in C} \delta_{|V|-1}(s, v') + \sum\limits_{v \in C} w(v', v) < \sum\limits_{v \in C} \delta_{|V|-1}(s, v)$
  - If C contains no witness, $\delta_{|V|}(s, v) \geq \delta_{|V|-1}(s, v)$ for all $v \in C$, a contradiction $\square$

Bellman-Ford

- **Idea!** Use **graph duplication**: make multiple copies (or levels) of the graph
- $|V|+1$ levels: vertex v_k in level k represents reaching vertex v from s using $\leq k$ edges
- If edges only increase in level, resulting graph is a DAG!

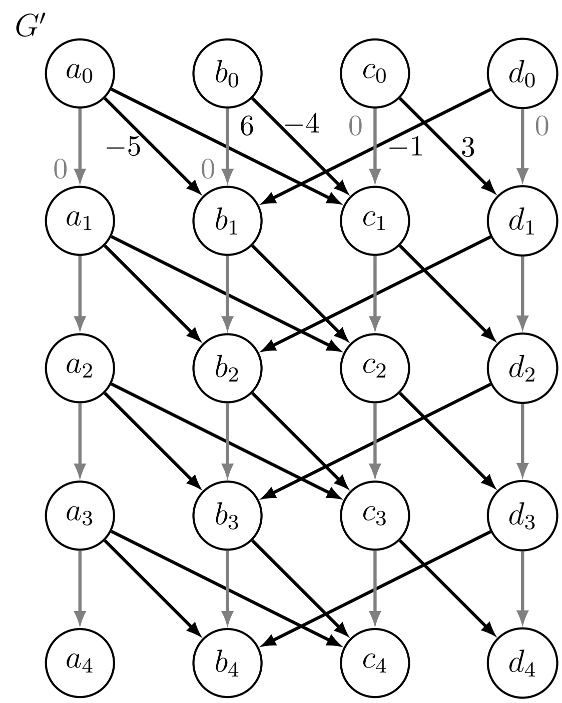
- Construct new DAG $G' = (V', E')$ from $G = (V, E)$:
  - G' has $|V|(|V|+1)$ vertices v_k for all $v \in V$ and $k \in \{0, \dots, |V|\}$
  - G' has $|V|(|V|+|E|)$ edges:
    * $|V|$ edges (v_{k-1}, v_k) for $k \in \{1, \dots, |V|\}$ of weight zero for each $v \in V$
    * $|V|$ edges (u_{k-1}, v_k) for $k \in \{1, \dots, |V|\}$ of weight $w(u, v)$ for each $(u, v) \in E$
- Run DAG Relaxation on G' from s_0 to compute $\delta(s_0, v_k)$ for all $v_k \in V'$
- For each vertex: set $d(s, v) = \delta(s_0, v_{|V|-1})$
- For each witness $u \in V$ where $\delta(s_0, u_{|V|}) < \delta(s_0, u_{|V|-1})$:
  - For each vertex v reachable from u in G:
    * set $d(s, v) = -\infty$

Example



$\delta(a_0, v_k)$

$k \setminus v$	a	b	c	d
0	0	∞	∞	∞
1	0	$\mathbf{-5}$	$\mathbf{6}$	∞
2	0	-5	$\mathbf{-9}$	$\mathbf{9}$
3	0	-5	-9	$\mathbf{-6}$
4	0	$\mathbf{-7}$	-9	-6
$\delta(a, v)$	0	$-\infty$	$-\infty$	$-\infty$



Correctness

- **Claim 3:** $\delta(s_0, v_k) = \delta_k(s, v)$ for all $v \in V$ and $k \in \{0, \ldots, |V|\}$
- **Proof:** By induction on k:
  - Base case: true for all $v \in V$ when $k = 0$ (only v_0 reachable from s_0 is $v = s$)
  - Inductive Step: Assume true for all $k < k'$, prove for $k = k'$

$$\begin{aligned}\delta(s_0, v_{k'}) &= \min\{\delta(s_0, u_{k'-1}) + w(u_{k'-1}, v_{k'}) \mid u_{k'-1} \in \text{Adj}^-(v_{k'})\} \\ &= \min\{\{\delta(s_0, u_{k'-1}) + w(u, v) \mid u \in \text{Adj}^-(v)\} \cup \{\delta(s_0, v_{k'-1})\}\} \\ &= \min\{\{\delta_{k'-1}(s, u) + w(u, v) \mid u \in \text{Adj}^-(v)\} \cup \{\delta_{k'-1}(s, v)\}\} \quad \text{(by induction)} \\ &= \delta_{k'}(s, v) \end{aligned}$$ □

- **Claim 4:** At the end of Bellman-Ford $d(s, v) = \delta(s, v)$
- **Proof:** Correctly computes $\delta_{|V|-1}(s, v)$ and $\delta_{|V|}(s, v)$ for all $v \in V$ by Claim 3
  - If $\delta(s, v) \neq -\infty$, correctly sets $d(s, v) = \delta_{|V|-1}(s, v) = \delta(s, v)$
  - Then sets $d(s, v) = -\infty$ for any v reachable from a witness; correct by Claim 2 □

Running Time

- G' has size $O(|V|(|V| + |E|))$ and can be constructed in as much time
- Running DAG Relaxation on G' takes linear time in the size of G'
- Does $O(1)$ work for each vertex reachable from a witness
- Finding reachability of a witness takes $O(|E|)$ time, with at most $O(|V|)$ witnesses: $O(|V||E|)$
- (Alternatively, connect **super node** x to witnesses via 0-weight edges, linear search from x)
- Pruning G at start to only subgraph reachable from s yields $O(|V||E|)$-time algorithm

Extras: Return Negative-Weight Cycle or Space Optimization

- **Claim 5:** Shortest $s_0 - v_{|V|}$ path π for any witness v contains a negative-weight cycle in G
- **Proof:** Since π contains $|V| + 1$ vertices, must contain at least one cycle C in G
  - C has negative weight (otherwise, remove C to make path π' with fewer vertices and $w(\pi') \leq w(\pi)$, contradicting witness v) □
- Can use just $O(|V|)$ space by storing only $\delta(s_0, v_{k-1})$ and $\delta(s_0, v_k)$ for each k from 1 to $|V|$
- Traditionally, Bellman-Ford stores only one value per vertex, attempting to relax every edge in $|V|$ rounds; but estimates do not correspond to k-Edge Distances, so analysis trickier
- But these space optimizations don't return a negative weight cycle