The Bellman-Ford Algorithm

The Basic Algorithm

Algorithms: Design and Analysis, Part II

The Recurrence

Notation: Let $L_{i,v}$ = minimum length of a s-v path with $\leq i$ edges.

- with cycles allowed
- defined as $+\infty$ if no s-v paths with $\leq i$ edges

Recurrence: For every $v \in V$, $i \in \{1, 2, 3, \ldots\}$

$$L_{i,v} = \min \begin{cases} L_{(i-1),v} & \text{Case 1} \\ \min_{(w,v) \in E} \{ L_{(i-1),w} + c_{wv} \} & \text{Case 2} \end{cases}$$

Correctness: brute-force search from the only (1+in-deg(v)) candidates (by the optimal substructure lemma).

If No Negative Cycles

Now: suppose input graph G has no negative cycles.

$\Rightarrow$ shortest paths do not have cycles

[removing a cycle only decreases length]

$\Rightarrow$ have $\leq (n-1)$ edges

Point: if G has no negative cycle, only need to solve subproblems up to $i = n-1$.

Subproblems: compute $L_{i,v}$ for all $i \in \{0,1,2,\ldots,n-1\}$ and all $v \in V$.

The Bellman-Ford Algorithm

Let A = 2-D array (indexed by i and v)

Base case: $A[0,s] = 0$; $A[0,v] = +\infty$ for all $v \neq s$

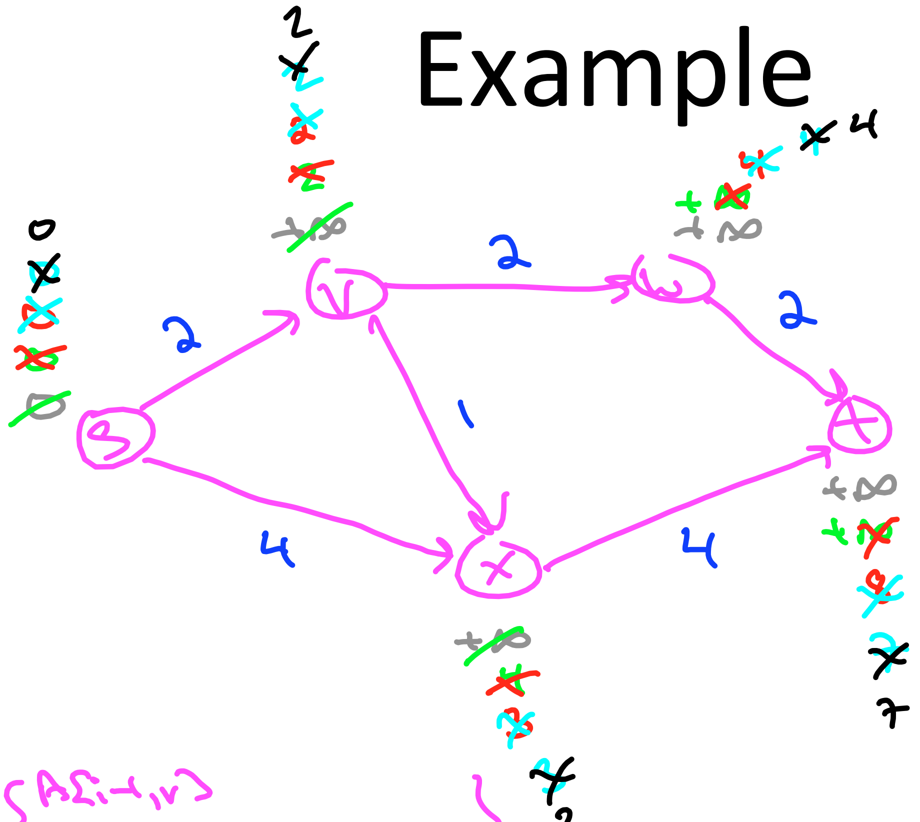
For $i = 1,2,3,\ldots,n-1$:

For each $v \in V$:

$$A[i,v] = \min \begin{cases} A[i-1,v] \\ \min_{(w,v)\in E} \{A[i-1,w] + c_{wv}\} \end{cases}$$

As discussed: if G has no negative cycle, then algorithm is correct [with final answers $= A[n-1,v]$'s]

Example



$i = 0$

$i = 1$

$i = 2$

$i = 3$

$i = 4$

$$A[i,v] = \min \begin{cases} A[i-1,v] \\ \min\limits_{(w,v)\in E} \{A[i-1,w] + c_{wv}\} \end{cases}$$

Quiz

Question: What is the running time of the Bellman-Ford algorithm? [Pick the strongest true statement.] [m = # of edges, n = # of vertices]

(A) $O(n^2)$ → # of subproblems, but might do $\Theta(n)$ work for one subproblem

(B) $O(mn)$

(C) $O(n^3)$

(D) $O(m^2)$

Reason:

Total work is $O(\underbrace{n}_{\substack{\text{\# iterations of outer loop} \\ \text{(i.e., choices of i)}}} \cdot \underbrace{\sum_{v \in V} \text{in-deg}(v)}_{\substack{\text{work done in one iteration} \\ = m}})$

$= O(mn)$

Stopping Early

Note: Suppose for some $j < n-1$, $A[j,v] = A[j-1,v]$ for all vertices v.

$\Rightarrow$ for all v, all future $A[i,v]$'s will be the same

$\Rightarrow$ can safely halt (since $A[n-1,v]$'s = correct shortest-path distances)