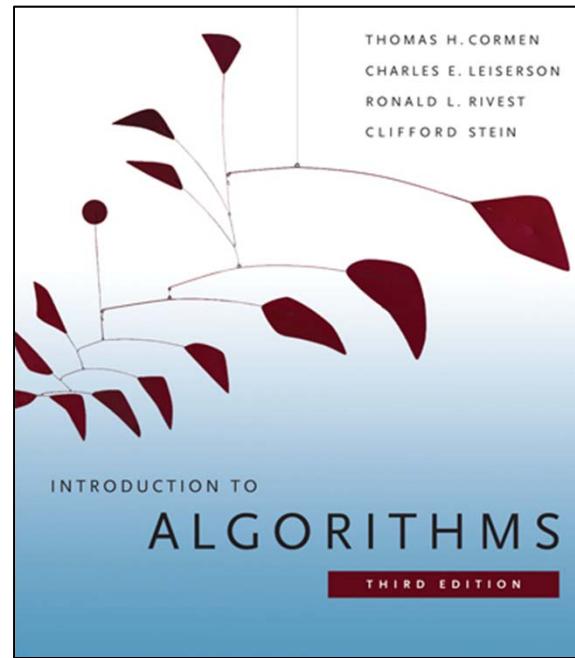


6.006

Introduction to Algorithms



Lecture 14: Shortest Paths I

Prof. Erik Demaine

Today

- Shortest paths
- Negative-weight cycles
- Triangle inequality
- Relaxation algorithm
- Optimal substructure

Shortest Paths

Google maps from: 32 Vassar Street, Cambridge, MA to:Times Square, New York, NY Search Maps

Get Directions My Maps

32 Vassar St, Cambridge, Middlesex, Massachusetts
Times Square, New York, NY Add Destination - Show options Get Directions

Driving directions to Theater District - Times Square, New York, NY

Suggested routes

- 1. I-84 W 3 hours 53 mins 221 mi
- 2. I-90 W 3 hours 58 mins 209 mi
- 3. I-395 S and I-95 S 4 hours 13 mins 227 mi

This route has tolls.

A 32 Vassar St Cambridge, MA 02139

1. Head southwest on Vassar St
2. Turn right at Memorial Dr
3. Turn left at Western Ave
4. Turn left at Soldiers Field Rd
5. Take the I-90 ramp Toll road
6. Keep right at the fork and merge onto I-90 W Partial toll road
7. Take exit 9 to merge onto I-84 W toward US-20/Hartford/New York City Partial toll road
8. Take exit 20 for I-684 toward NY-22/White Plains/Pawling
9. Keep left at the fork and merge onto I-684 S
10. Merge onto Hutchinson River Pkwy
11. Continue onto Cross County Pkwy (signs for George Washington Bridge)

Print Send eLink Satellite Traffic

©2011 Google - Map data ©2011 Google - Terms of Use Report a problem

Shortest Paths

oracleofbacon.org/cgi-bin/movielinks

The Oracle of Bacon website displays a shortest path from Kevin Bacon to Dr. Erik D. Demaine. The path consists of the following nodes and edges:

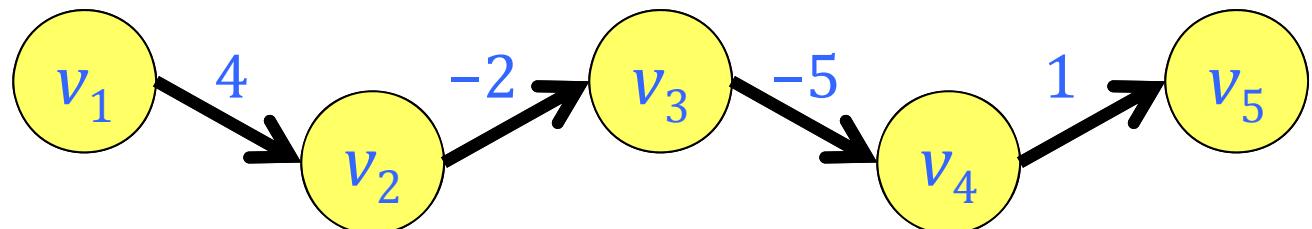
- Kevin Bacon (green box)
- was in |
The Man Who Saved Geometry (2009) (TV) (purple box)
with
Benoît B. Mandelbrot (green box)
- was in |
The Revenge of the Dead Indians (1993) (purple box)
with
Dennis Hopper (green box)
- was in |
The 2004 IFP/West Independent Spirit Awards (2004) (TV) (purple box)
with
Kevin Bacon (green box)

Kevin Bacon to Dr. Erik D. Demaine Find link More options >>

How Long Is Your Path?

- Directed graph $G = (V, E)$
- Edge-weight function $w : E \rightarrow \mathbb{R}$
- Path $p = v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k$
- **Weight** of p , denoted $w(p)$, is
 $w(v_1, v_2) + w(v_2, v_3) + \dots + w(v_{k-1}, v_k)$

Example:

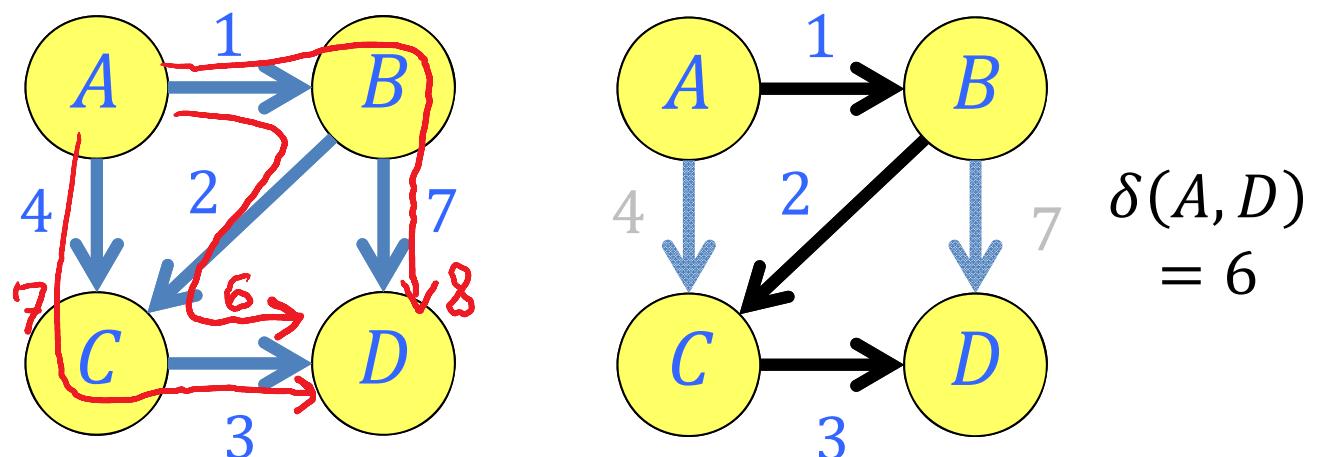


$$w(p) = 4 - 2 - 5 + 1 = -2$$

My Path Is Shorter Than Yours

- A *shortest path* from u to v is a path p of minimum possible weight $w(p)$ from u to v
- The shortest-path weight $\delta(u, v)$ from u to v is the weight of any such shortest path:
$$\delta(u, v) = \min \{w(p) : p \text{ is a path from } u \text{ to } v\}$$

Example:



You Can't Get There From Here

Google maps from 32 Vassar Street, Cambridge, MA to:tokyo japan

Get Directions My Maps

36. Take exit 13B toward Halawa Hts. Stadium 0.3 mi

37. Merge onto I-H-201 E 4.1 mi

38. Merge onto I-H-1 E 4.1 mi

39. Take exit 23 for Punahou St toward Waikiki/Manoa 0.2 mi

40. Turn right at Punahou St 0.1 mi

41. Take the 1st right onto S Beretania St 0.1 mi

42. Take the 1st left onto Kalakaua Ave 0.1 mi

**43. Kayak across the Pacific Ocean
Entering Japan** 3,879 mi

44. Turn left toward 県道275号線 0.4 mi

45. Turn left toward 県道275号線 358 ft

46. Turn left toward 県道275号線 0.2 mi

47. Turn right at 県道275号線 0.1 mi

48. Turn left at 国道125号線 499 ft

49. Turn right at 国道24号線 0.6 mi

50. Turn left at 千束町(交差点) onto 国道354号線 0.6 mi

51. Turn right at 中村陸橋下(交差点) to stay on 国道354号線 1.0 mi

52. Take the ramp to 常磐自動車道 Toll road 0.3 mi

53. Keep left at the fork, follow signs for 東京 and merge onto 常磐自動車道 Toll road 23.8 mi

54. Take exit 三郷J C T on the right toward 首都高速・銀座・渋谷 0.3 mi

Print Send Link

Satellite

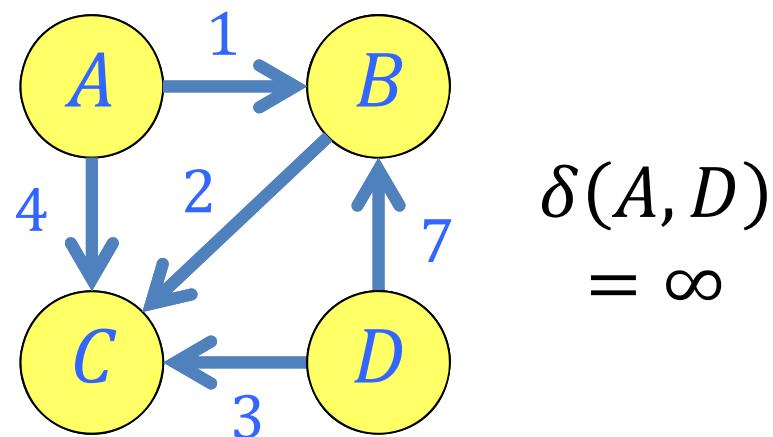
Traffic

©2011 Google - Map data ©2011 Europa Technologies, Geocentre Consulting, INEGI, MapLink, Tele Atlas, Whereis(R), Sensis Pty Ltd - Terms of Use

You Can't Get There From Here

- If there is no path from u to v , then neither is there a *shortest* path from u to v
- Define $\delta(u, v) = \infty$ in this case

Example:



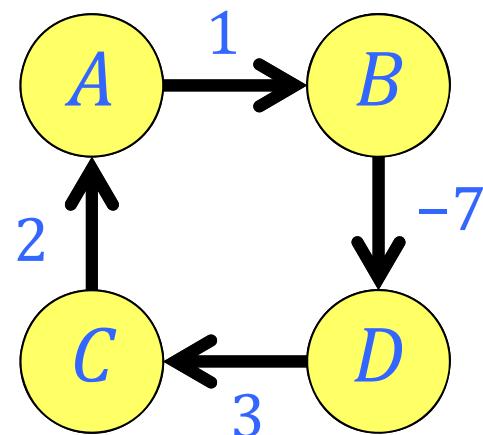
The More I Walk, The Less It Takes

- A shortest path from u to v might not exist, even though there is a path from u to v
- *Negative-weight cycle*

$$c = v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_1$$

has $w(c) < 0$

Example:

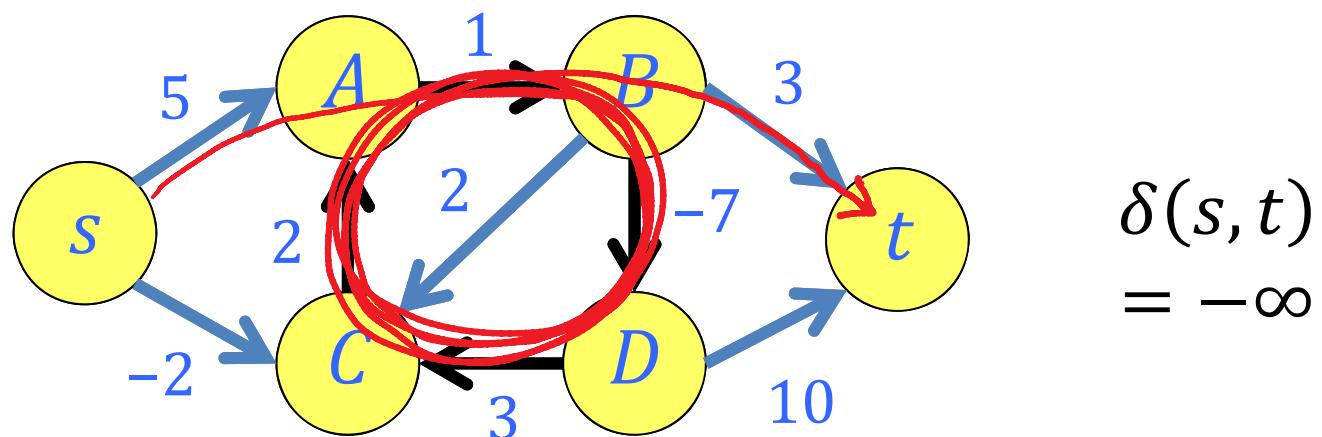


$$\begin{aligned} w(A \rightarrow B \rightarrow C \\ \rightarrow D \rightarrow A) \\ = -1 \end{aligned}$$

The More I Walk, The Less It Takes

- Define $\delta(u, v) = -\infty$ if there's a path from u to v that visits a negative-weight cycle
- $\delta(u, v) = \inf \{w(p) : p \text{ is a path from } u \text{ to } v\}$

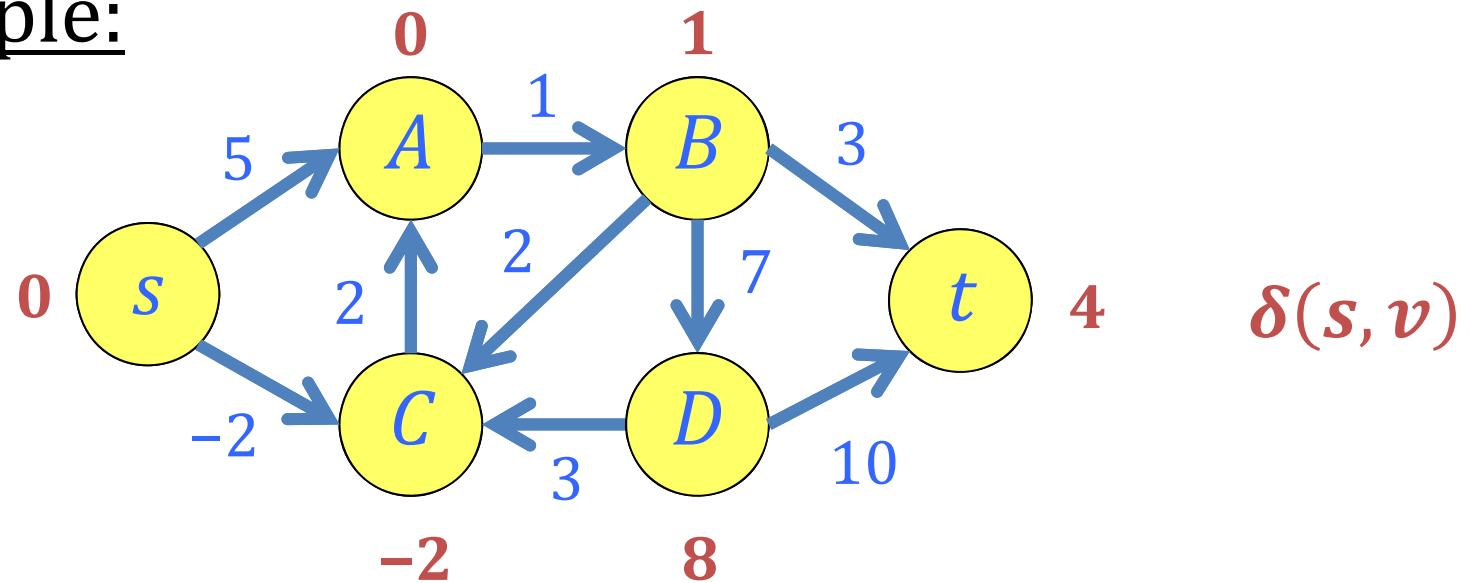
Example:



Single-Source Shortest Paths

- Problem: Given a directed graph $G = (V, E)$ with edge-weight function $w : E \rightarrow \mathbb{R}$, and a *source* vertex s , compute $\delta(s, v)$ for all $v \in V$

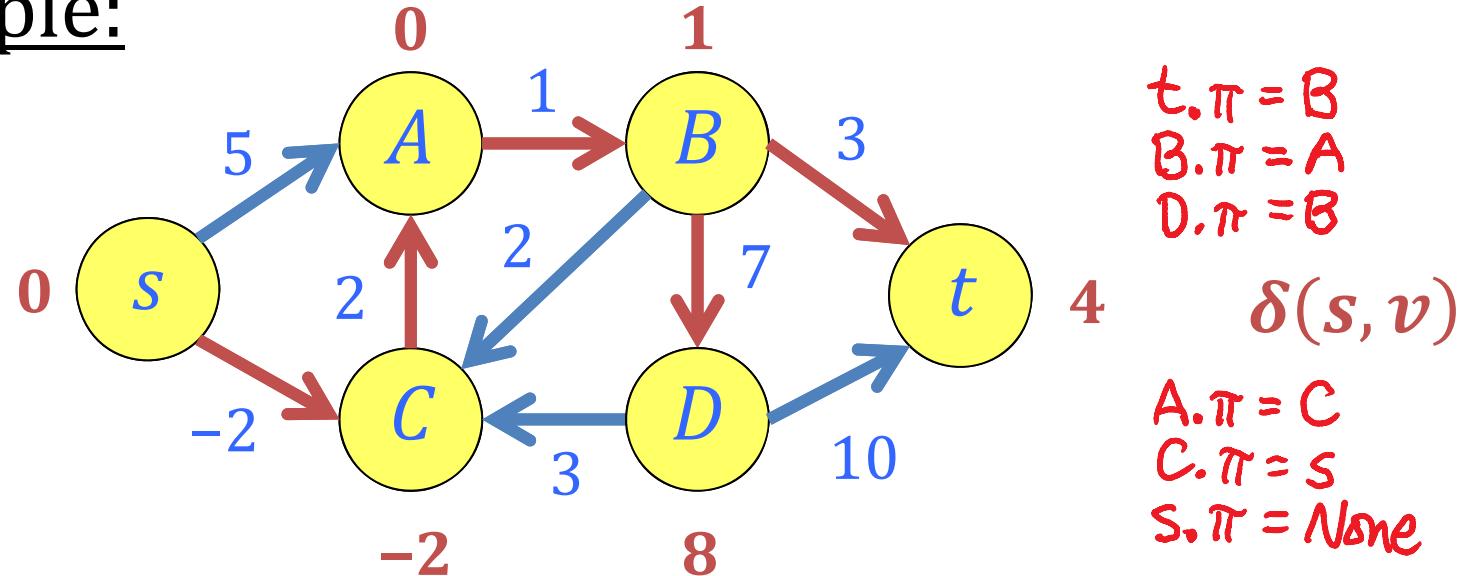
Example:



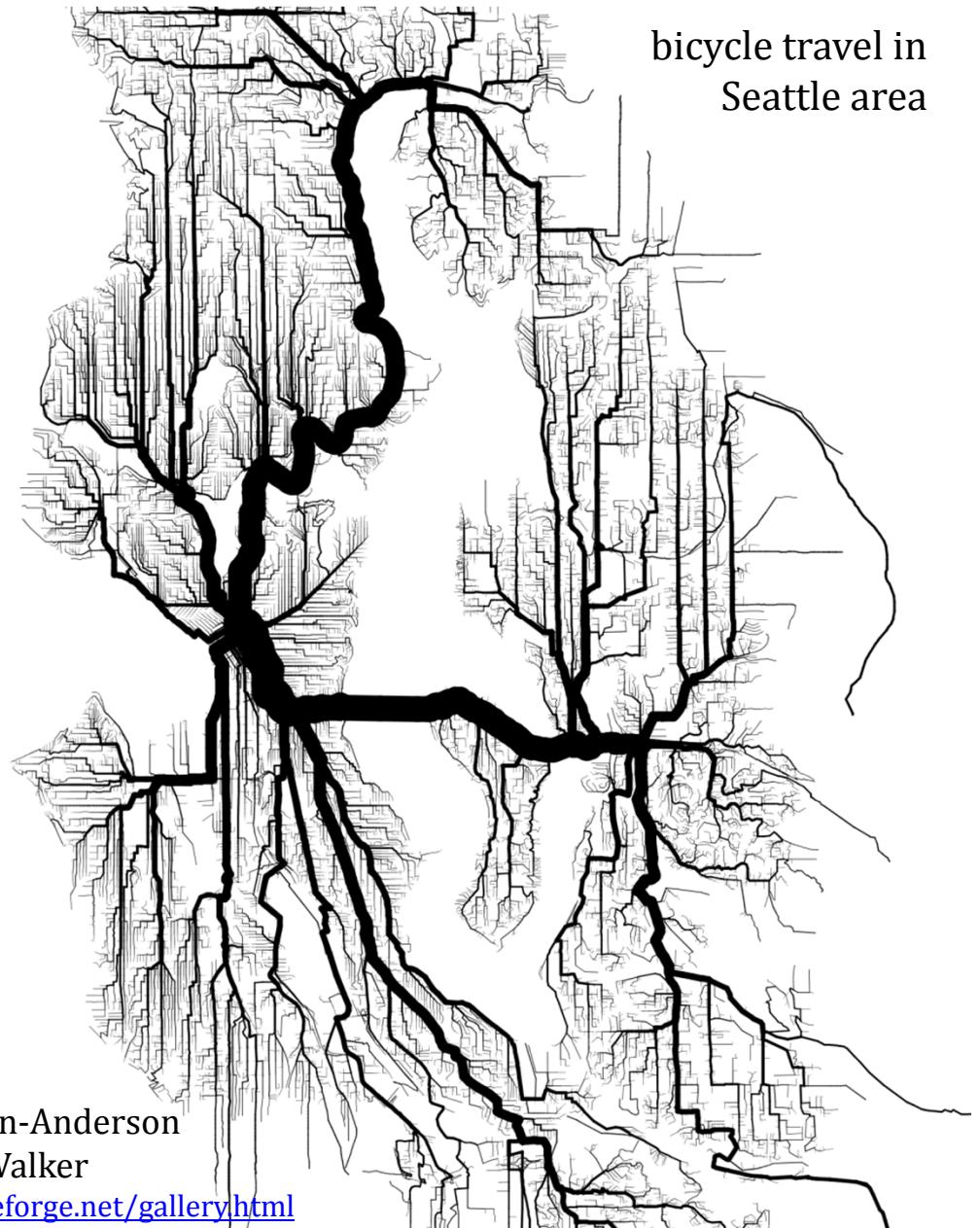
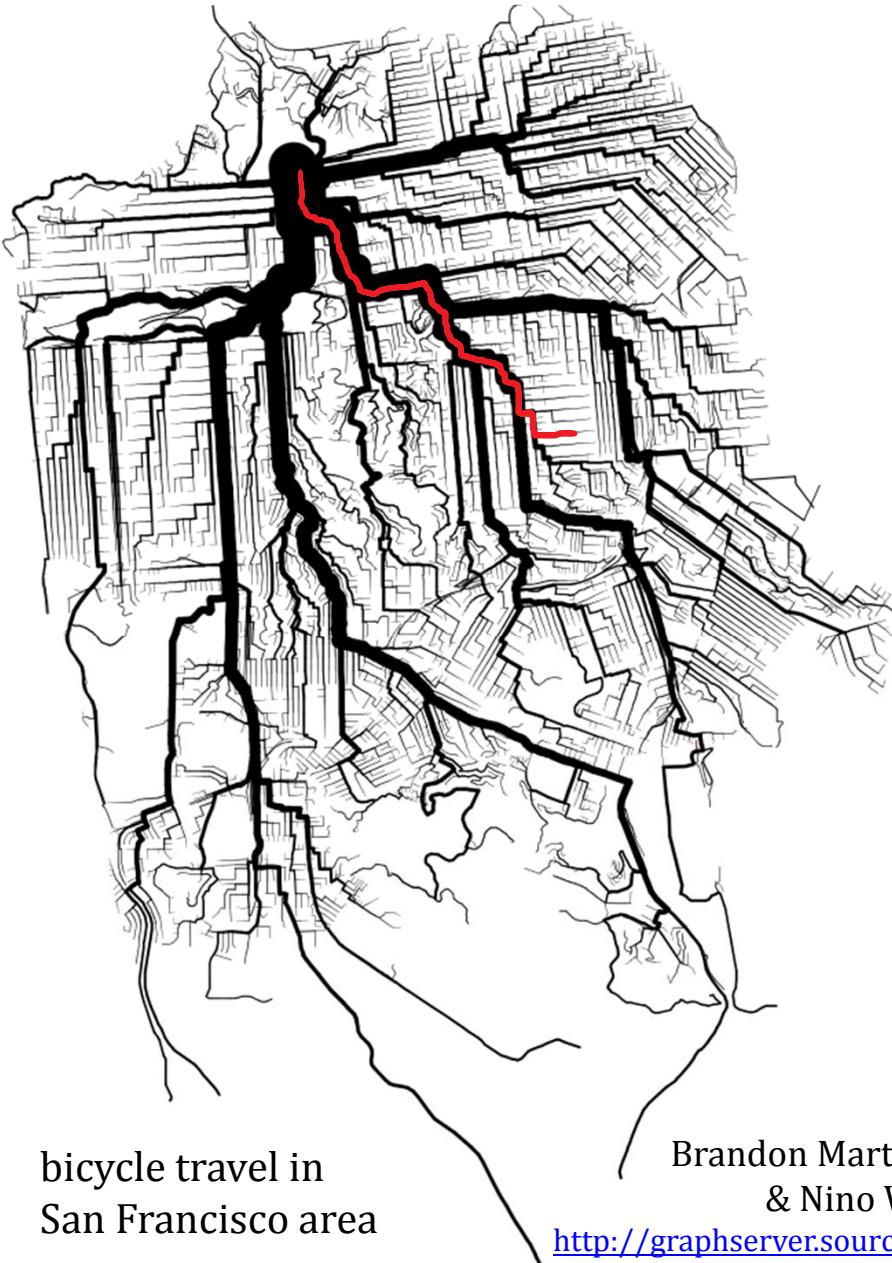
Shortest-Path Tree

- Ideally also compute a ***shortest-path tree*** containing a shortest path from source s to every $v \in V$ (assuming shortest paths exist)
 - Represent by storing ***parent*** $v.\pi$ for each $v \in V$
 $= \pi[v]$ in earlier CLR(s)

Example:



Shortest-Path Trees



Brandon Martin-Anderson
& Nino Walker

<http://graphserver.sourceforge.net/galleryhtml>

Single-Source Shortest-Path Algorithms

- **Relaxation algorithm** *(TODAY)*
 - Framework for most shortest-path algorithms
 - Not necessarily efficient
- **Bellman-Ford algorithm** *(LECTURE 15)*
 - Deals with negative weights
 - Slow but polynomial
- **Dijkstra's algorithm** *(LECTURE 16)*
 - Fast (nearly linear time)
 - Requires nonnegative weights

Brute-Force Algorithm

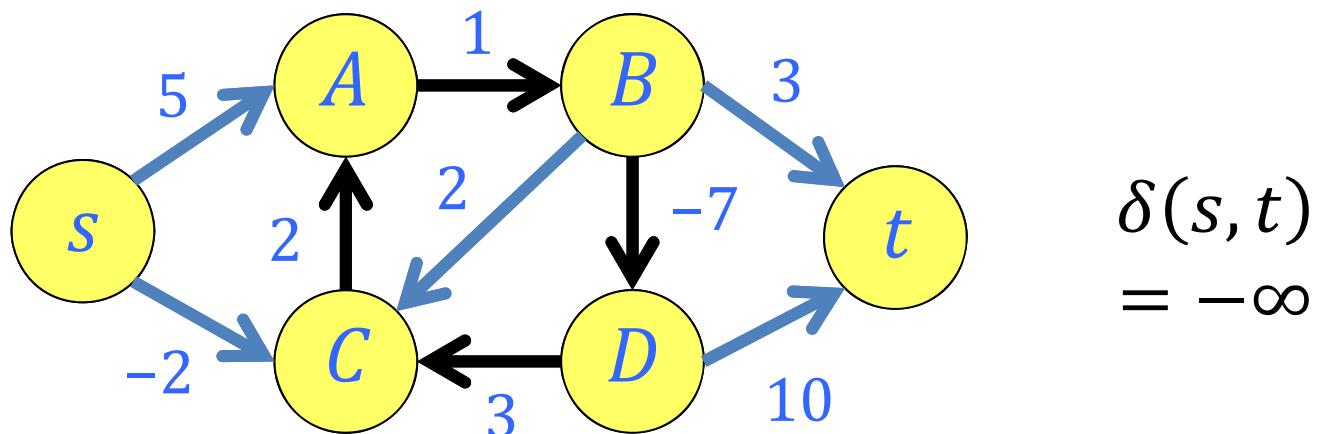
$\text{distance}(s, t)$:

for each path p from s to t :

compute $w(p)$

return p encountered with smallest $w(p)$

- Number of paths can be *infinite*:



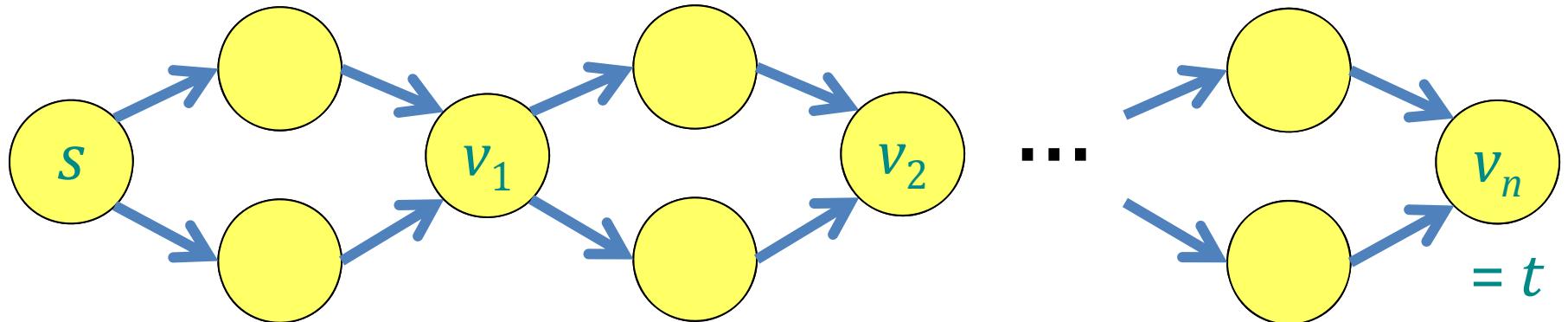


$$(n-2)! \sim n^n = 2^{n \lg n}$$

Brute-Force Algorithm

```
distance( $s, t$ ): ## assume no negative-weight cycles  
    for each simple path  $p$  from  $s$  to  $t$ :  $\leftarrow$  # paths  
        compute  $w(p)$   $\leftarrow O(v)$   
    return  $p$  encountered with smallest  $w(p)$ 
```

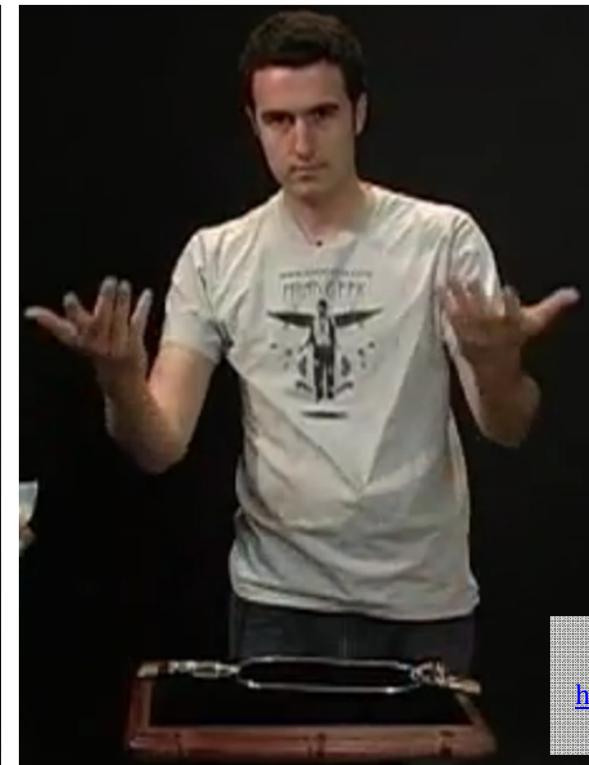
- Number of paths can be *exponential*:



2^n paths from s to v_n ; $O(n)$ vertices and edges

Relaxation

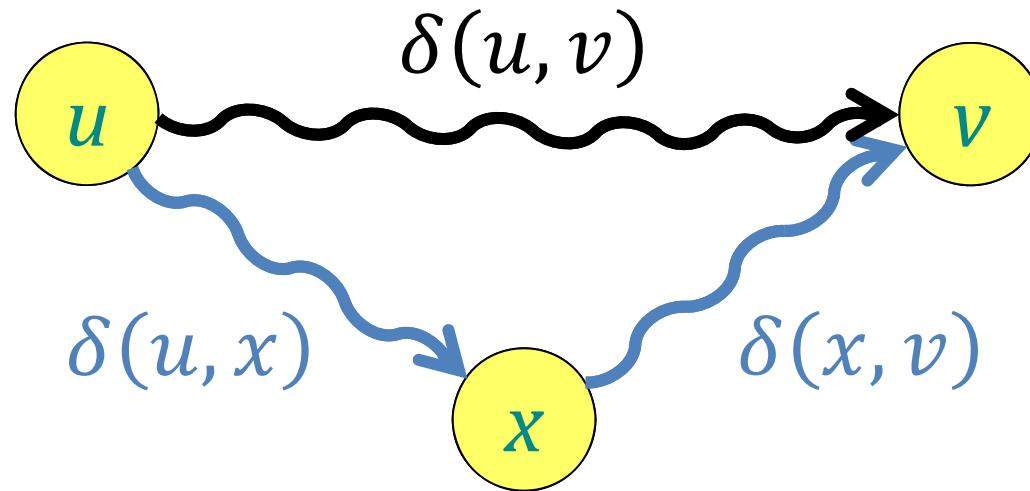
- In general, refers to letting a solution (temporarily) violate a constraint, and trying to fix these violations



Magic Geek
[http://www.youtube.com/
watch?v=Y12daEZTUYo](http://www.youtube.com/watch?v=Y12daEZTUYo)

Triangle Inequality

- Theorem: For all $u, v, x \in V$, we have
$$\delta(u, v) \leq \delta(u, x) + \delta(x, v).$$



- Proof: Shortest path from u to v is at most any particular path, e.g., the blue chain. ■

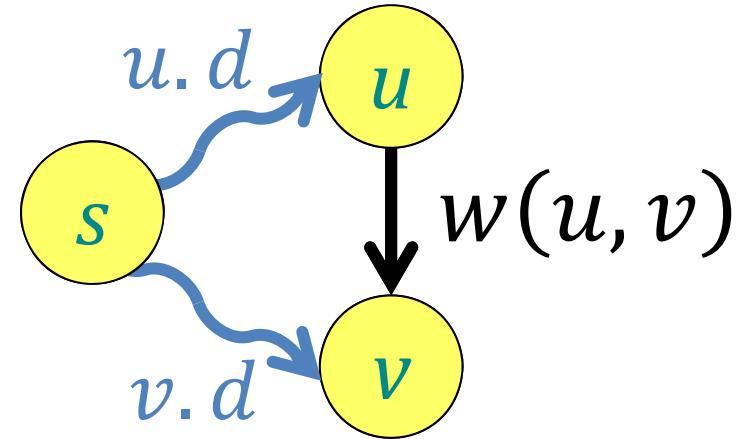
Relaxation Approach

- Maintain ***distance estimate*** $v.d = d[v]$ in older CLR(s) for each $v \in V$
- Goal: $v.d = \delta(s, v)$ for all $v \in V$
- Invariant: $v.d \geq \delta(s, v)$
- Initialization:

```
for v in V:  
    v.d = infinity  
s.d = 0
```
- Repeatedly improve estimates toward goal, by aiming to achieve triangle inequality

Edge Relaxation

- Consider an edge (u, v)



- $\delta(s, v) \leq \delta(s, u) + \delta(u, v)$ [triangle ineq.]
 $\leq \delta(s, u) + w(u, v)$ [candidate path]

\Rightarrow want $v.d \leq u.d + w(u, v)$

relax(u, v):

if $v.d > u.d + w(u, v)$:

$v.d = u.d + w(u, v)$

Relaxation Algorithm

for v in V :

$$v.d = \infty$$

$$s.d = 0$$

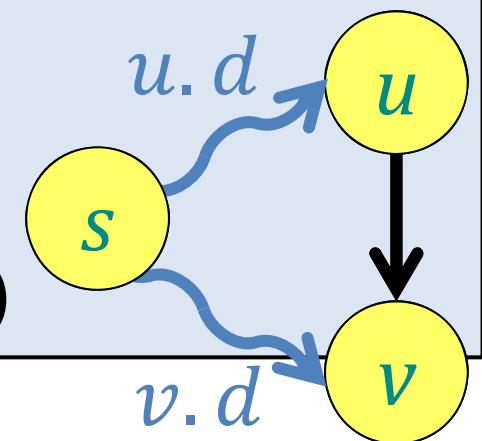
while some edge (u, v) has $v.d > u.d + w(u, v)$:

pick such an edge (u, v)

relax (u, v) :

if $v.d > u.d + w(u, v)$:

$$v.d = u.d + w(u, v)$$



Relaxation Algorithm with Shortest-Path Tree

for v in V :

$$v.d = \infty$$

$$v.\pi = \text{None}$$

$$s.d = 0$$

while some edge (u, v) has $v.d > u.d + w(u, v)$:

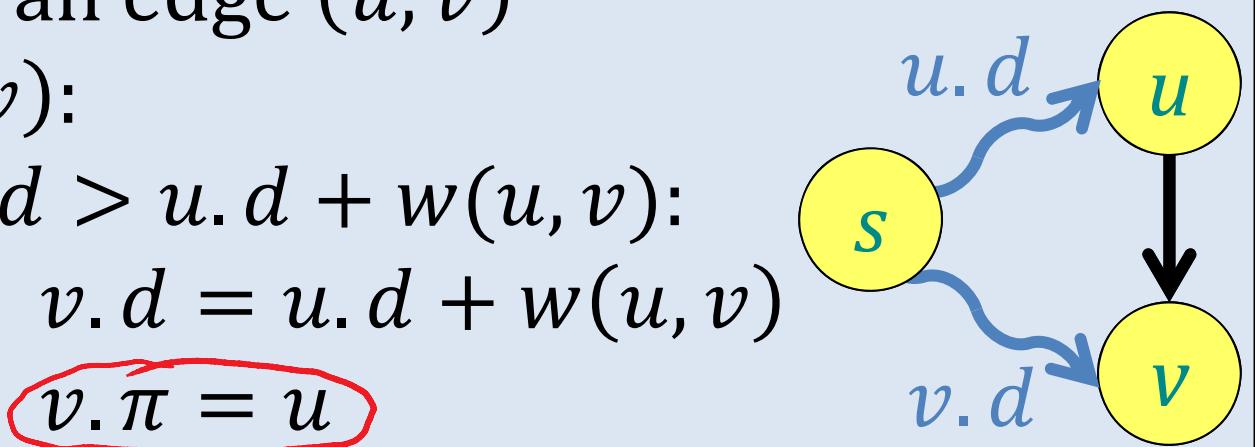
pick such an edge (u, v)

relax (u, v) :

if $v.d > u.d + w(u, v)$:

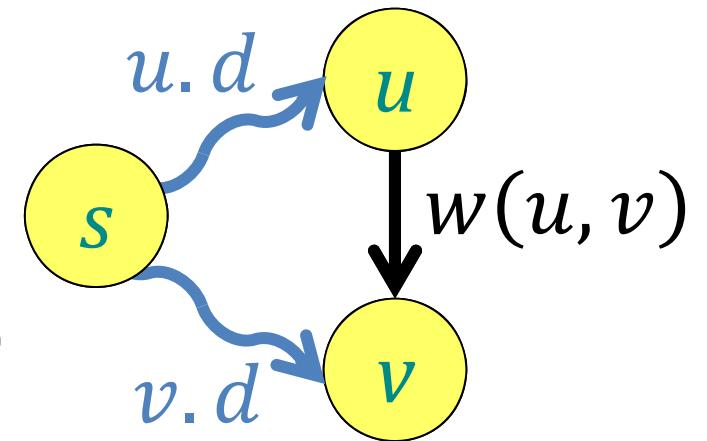
$$v.d = u.d + w(u, v)$$

$$v.\pi = u$$



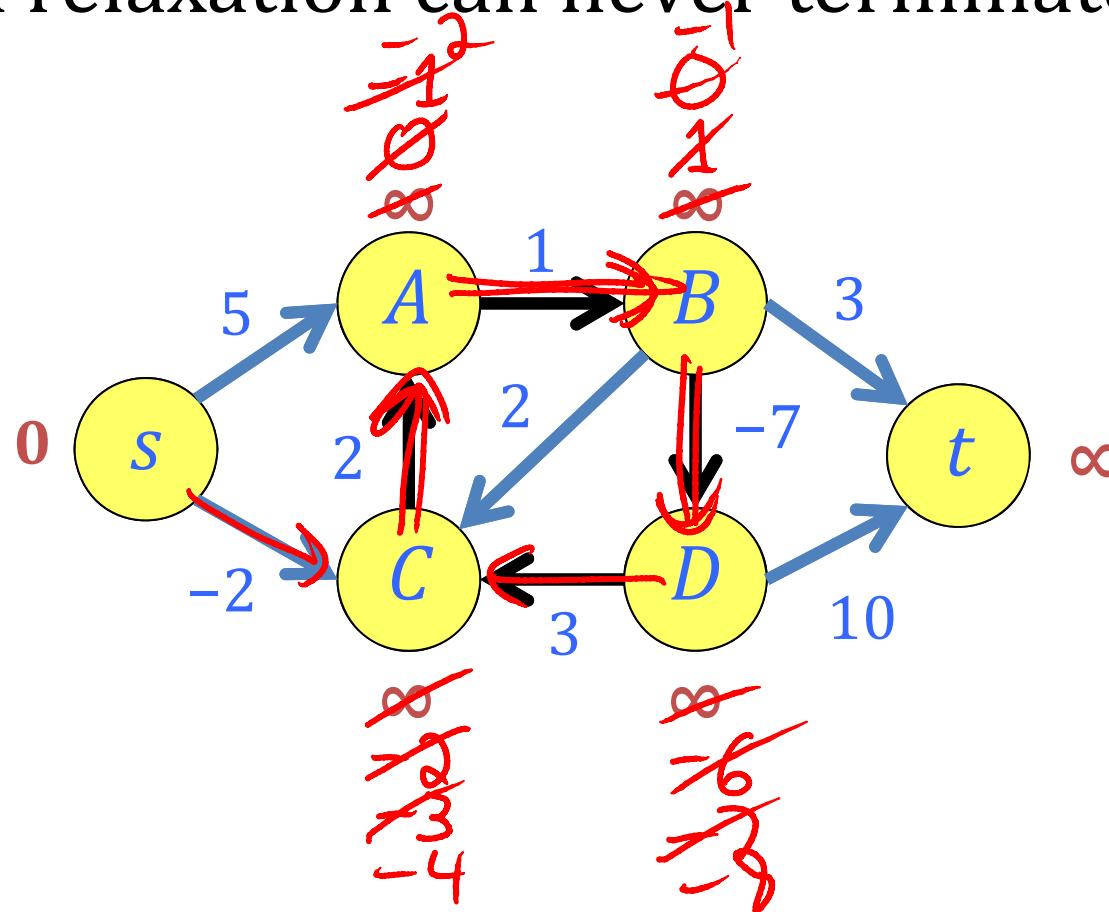
Relaxing Is Safe

- Lemma: The relaxation algorithm maintains the invariant that $v.d \geq \delta(s, v)$ for all $v \in V$.
- Proof: By induction on the number of steps.
 - Consider $\text{relax}(u, v)$
 - By induction, $u.d \geq \delta(s, u)$
 - By triangle inequality,
$$\begin{aligned}\delta(s, v) &\leq \delta(s, u) + \delta(u, v) \\ &\leq u.d + w(u, v)\end{aligned}$$
 - So setting $v.d = u.d + w(u, v)$ is “safe” ■

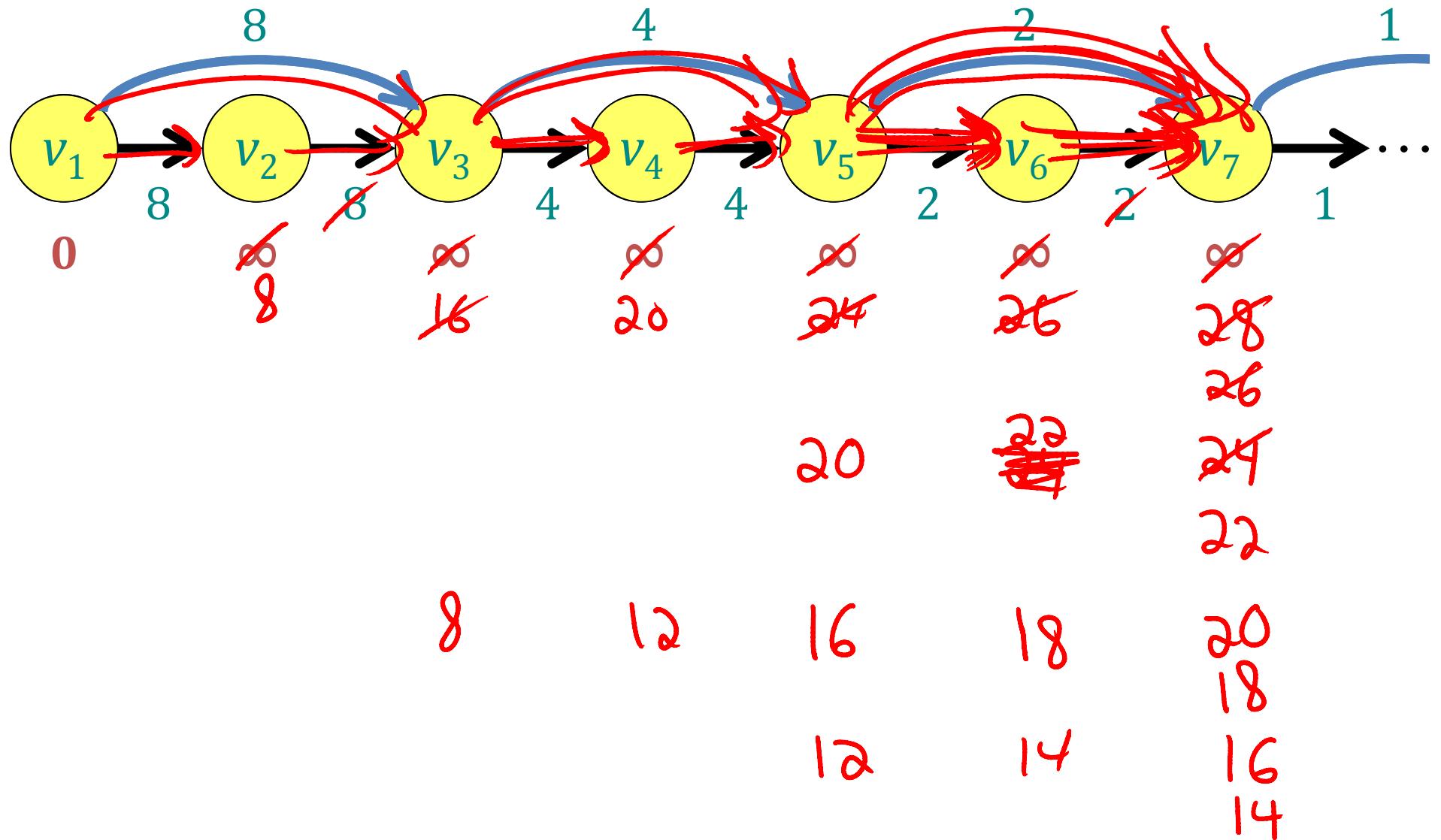


Infinite Relaxation

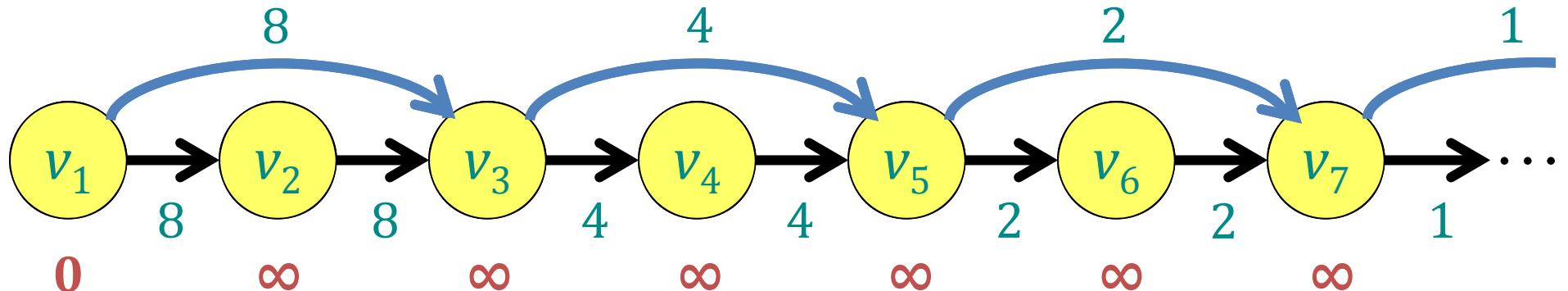
- If a negative-weight cycle is reachable from s , then relaxation can never terminate



Long Relaxation



Long Relaxation



- Analysis:

- $\text{relax}(v_1, v_2)$ - 1

$$T(n) = 2 T(n - 2) + 3$$

- $\text{relax}(v_2, v_3)$ - 2

$$T(n) = \Theta(2^{n/2})$$

- recurse on v_3, v_4, \dots, v_n } $T(n-2)$

- $\text{relax}(v_1, v_3)$ - 3

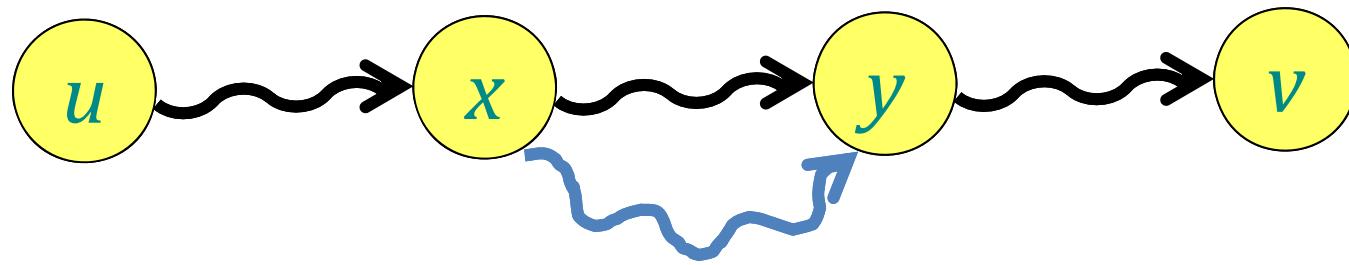
- recurse on v_3, v_4, \dots, v_n } $T(n-2)$

Are You Sure This Is a Good Idea?

- **Bellman-Ford algorithm:** *(LECTURE 15)*
 - Relax all of the edges
 - Repeat $\sim |V|$ times
 - Polynomial time!
- **Dijkstra's algorithm:** *(LECTURE 16)*
 - Relax edges in a growing ball around s
 - Nearly linear time!
 - (but doesn't work with negative edge weights)

Optimal Substructure

- Lemma: A subpath of a shortest path is a shortest path (between its endpoints).



- Proof: By contradiction.
 - If there were a shorter path from x to y , then we could **shortcut** the path from u to v , contradicting that we had a shortest path. ■