CS3000
6/9- Mon.

Admin

- HW5 due $\boxed{6/12}$ 9pm
  - ↳ submit on time if want for 2nd chance
- APP8 due 11:30AM tmw
- practice for exam 2
  - ↳ solns out tmw
  - review in 6/10 recitation
- XC exam 8 on 6/18

Agenda

1. Single Source Shortest Paths
2. Dijkstra's Overview
3. Dijkstra's Proof
4. APP8

$\boxed{1.\ \text{SSSP}}$

↳ from a starting vertex s

shortest path from s to every reachable vertex

shortest path $s \rightsquigarrow v$ is a path from s to v with least total weight

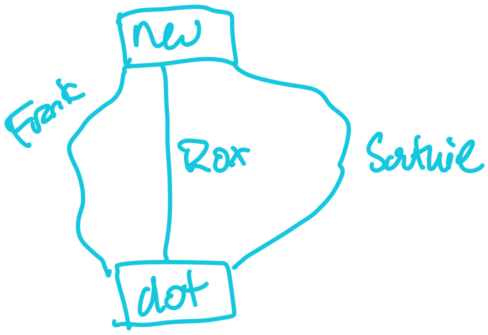
today's graphs:
- directed
- weighted

so far...

- BFS: shortest path: # edges on the path ☆
  undirected, unweighted
- DFS: all reachable vertices
  directed, unweighted
  topo sort, cycle detection
- MST: least total weight for a spanning tree (subset of E, all V) ☆
  undirected, weighted

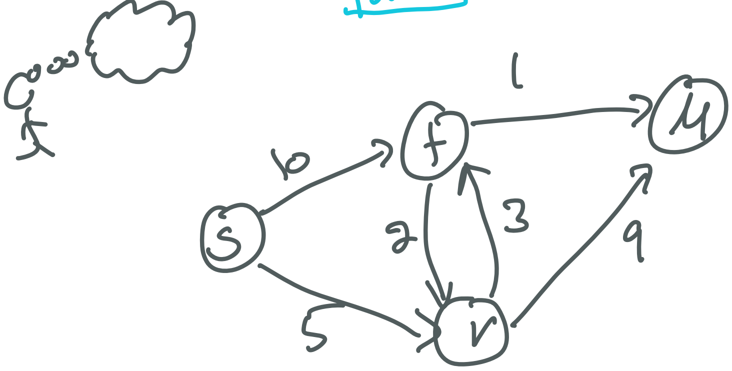
optimization

today's approach



- optimize: overall weight (not # edges)
- IRL: weights represent time, distance

(ex)



valid path from s to t

s, t	w = 10
s, v, t	w = 5 + 3 = 8
s, v, t, v, t	w = 13

optimizing for shortest path

SSSP — Dijkstra's, Bellman-Ford

needed for Dijkstra's:
- weighted
- cycles ok!
- non negative weights
- directed?

2. Dijkstra's overview

↳ greedy soln for SSSP

- like Prim's – start with path to every vertex as we explore it
  but! path might get update if we can get to the vertex another way

vertex attributes:

$v.d$ = distance from $s \rightsquigarrow v$

$v.\pi$ = predecessor on shortest path

initialize:

$v.d = \infty$ $\quad s.d = 0$

$v.\pi = nil$ $\quad s.\pi = nil$

Algo: DIJKSTRA(G, w, s)

Graph adj list — weight $w(u,v)$ — starting vertex

First guess:

best way to get to s's neighbors is directly from s

Relaxing

explore other vertices, possibly update path/distance

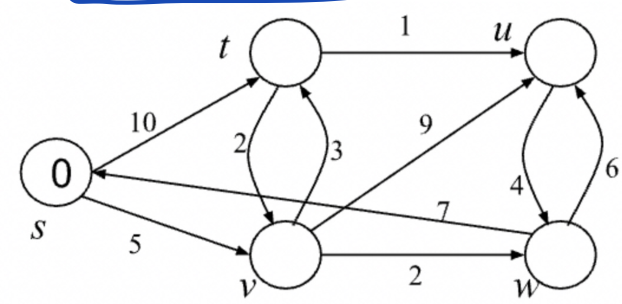
RELAX(u, v, w) ⤳ is it better to get to v by going through u? If so, update v

$$\boxed{\begin{array}{l}\text{if } u.d + w(u,v) < v.d \\ \quad v.d = u.d + w(u,v) \\ \quad v.\pi = u\end{array}}$$

Also needed:

Heap H
- vertices to proc
- keyed on v.d

Set A
- vertices in proc or done
- v.d is final!



H	extracted	A	vertex updates
s, t, v, u, w	s	{s}	t.d = 10 v.d = 5 t.π, v.π = s
t, v, u, w	v	{s, v}	t.d = 8 w.d = 7 u.d = 14 t.π, w.π, u.π = v
t, u, w	w	{s, v, w}	u.d = 13 u.π = w
t, u	t	{s, v, w, t}	u.d = 9 u.π = t
u	u	{s, v, w, t, u}	none!

3. Dijkstra's Proof

⤷ value of optimal solution (distance values)

Goal: $\boxed{v.d = \delta(s,v) \text{ for all } v \in V}$

$\delta(s,v)$ — distance of actual shortest path from s to v

$v.d$ — distance value produced by the algorithm

Agree on:

#1 $v.d \geq \delta(s,v)$
- if v.d is wrong, it's worse and not better than optimal

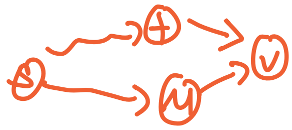
#2 actual shortest path $s \rightsquigarrow v$, (u, v) is edge in the path

$s \rightsquigarrow u \rightarrow v$

If $u.d = \delta(s,u)$, then $v.d = \delta(s,v)$

after RELAX edge (u,v)

what if ...



but this is actual s.p.!

now ... proof by induction on $|A|$

- vertex is added to A, distance value is finalized
- before adding to A, vertex is removed from Heap H

Base case

$|A| = 1$ when $|A| = 1$, the only vertex in it is s. $s.d = 0$

$\delta(s,s) = 0$ ✓

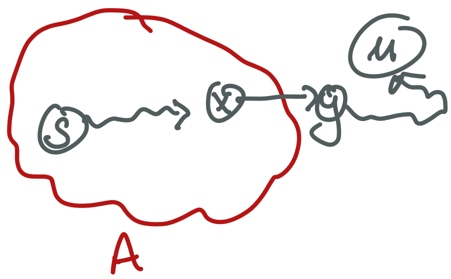
IH ★ $v.d = \delta(s,v)$ for every $v \in A$ up to $|A| = k$

↳ now we add vertex u, the $k{+}1^{th}$ vertex into A

- $u.d$ is final
- $u.d$ was smallest in heap
- at some point, we relaxed (x,y)

↳ by IH, $x.d = \delta(s,x)$

by #2, relaxing (x,y) gives us $y.d = \delta(s,y)$



actual s.p. from s to u

y precedes u on s.p.

edge (x,y)

x is in A

Everything we know!

$x.d = \delta(s,x)$ by IH

$\delta(s,y) \le \delta(s,u)$ b/c y precedes u

non-neg edge weights

$u.d \le y.d$ b/c u was extracted first

$u.d \ge \delta(s,u)$ by #1

$y.d = \delta(s,y)$ by #2

string inequalities together

$\delta(s,y) \le \delta(s,u) \le u.d \le y.d$

but! $\delta(s,y) = y.d$

soo....

$\delta(s,y) = \delta(s,u) = u.d = y.d$

↳ $u.d = \delta(s,u)$!!! ★ ★ ★