Math 182 Lecture 14

§8.2 Breadth-first search (BFS)

Given graph $G=(V,E)$
given $s \in V$ called the source.
BFS will seach all vertices which are "reachable" from s,
(v reachable from s if $\exists$ path $s \rightsquigarrow v$)

BFS will do:

(1) computes shortest-path distance from s to all other vertices.

(2) creates a breadth-first tree which contains shortest-length paths from s to other vertices on the tree. This tree contains all reachable vertices from s.

In BFS, vertices will have a color:

v.color
- White → means vertex not yet discovered
- Gray → vertex discovered, in the frontier
- Black → vertex discovered, already had neighbors processed.

distance
$v.d$ = best known path-distance from s.

predecessor:
$v.\pi$ = vertex from which we discovered v.

BFS(G, s)

```
for each vertex u ∈ V, V \ {s}
    u.color = WHITE
    u.d = ∞
    u.π = NIL
s.color = GRAY
s.d = 0
s.π = NIL
Q = ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    u = DEQUEUE(Q)
    for each v ∈ G.Adj[u]
        if v.color == WHITE
            v.color = GRAY
            v.d = u.d + 1
            v.π = u
            ENQUEUE(Q, v)
    u.color = BLACK
```

initialize
$u \in V \setminus \{s\}$
u.color = white
$u.d = \infty$
$u.\pi$ = nil

initialize s

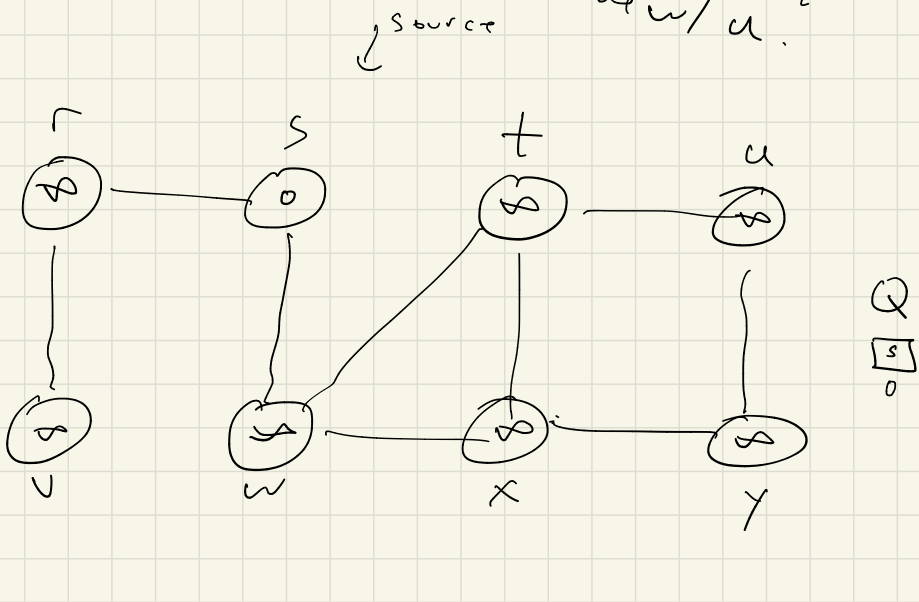
create queue Q

put s in Q.

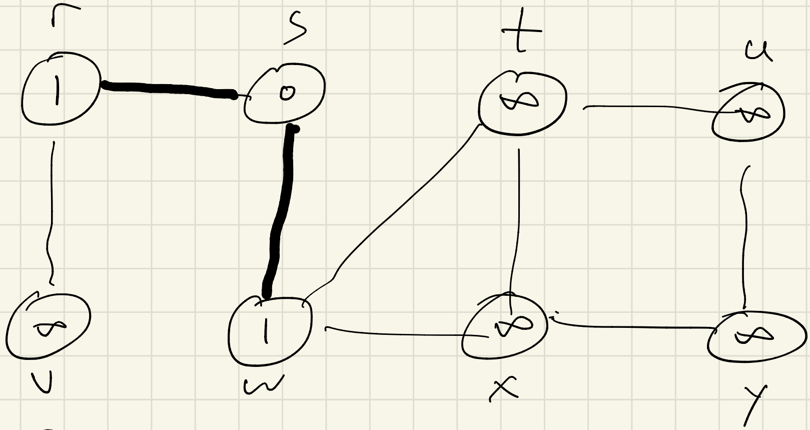
if neighbor v is undiscovered (white)
Then discover it
and define color, distance and π

put v in back of queue

done w/ u.



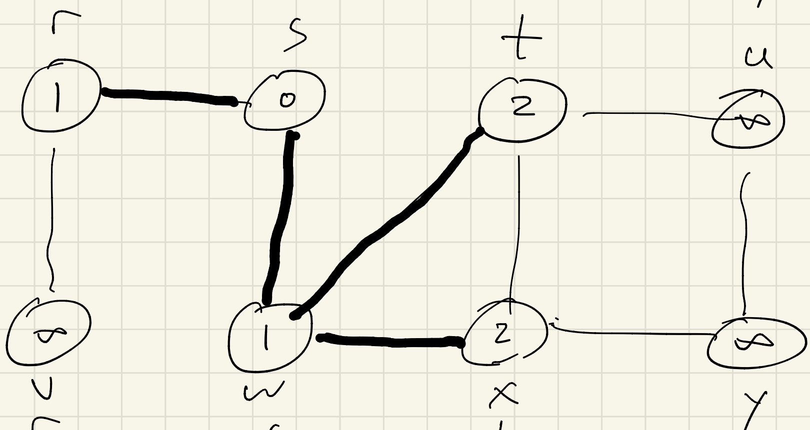
u=s



Q

w	r
1	1

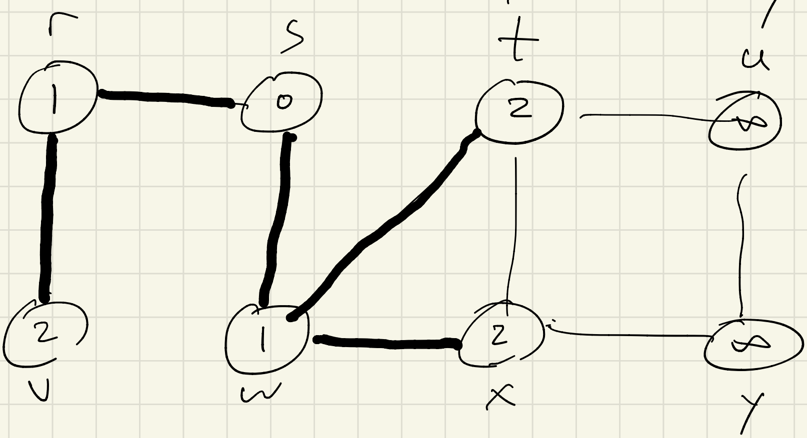
u=w



Q

~~r~~	t	x
1	2	2

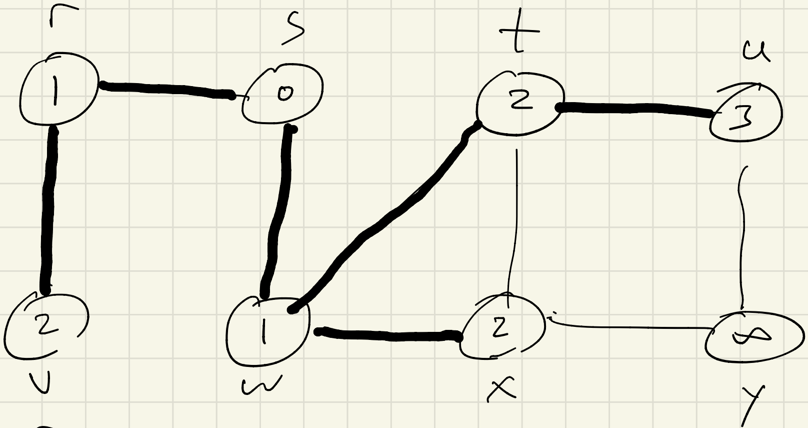
u=r



Q

t	x	v
2	2	2

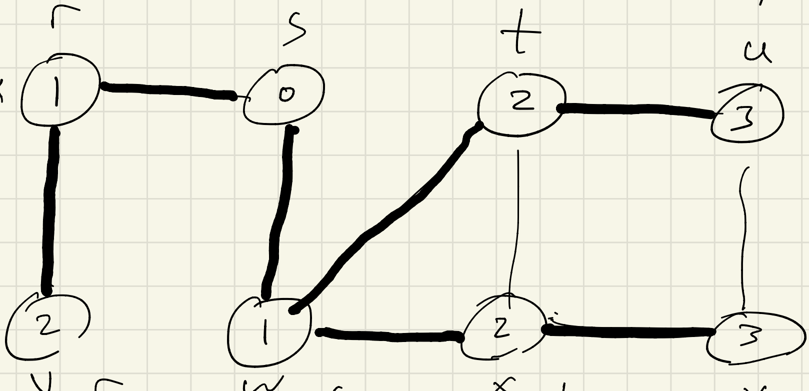
$u = t$



Q

x	v	u
2	2	3

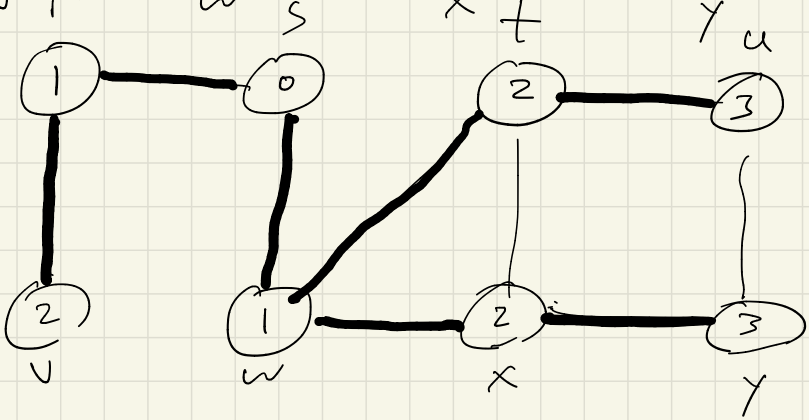
$u = x$



Q

v	u	y
2	3	3

$u = v$
$u = u$
$u = y$



Q

Running-time analysis

lines 1-9 initialization $\Theta(V)$

while loop runs $O(V)$ times,
for loop runs $O(E)$ times

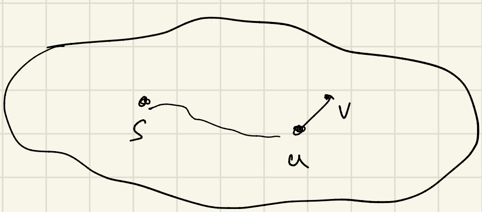
Overall BFS $O(V+E)$.

Shortest paths Defs

Given $G=(V,E)$ $s \in V$ source

define <u>shortest-path distance</u> $\delta(s,v)$ from s to v to be minimum # of edges in some path from s to v, or $\delta(s,v)=\infty$ if no path exists.

Call path from s to v a <u>shortest path</u> if length of path $= \delta(s,v)$.



Lemma 8.2.1. *Let $G = (V, E)$ be a directed or undirected graph, and let $s \in V$ be an arbitrary vertex. Then, for any edge $(u, v) \in E$,*

$$\delta(s, v) \leq \delta(s, u) + 1.$$

<u>Proof</u>: Sps $\delta(s,v) = \infty$, s v not reachable from s, so u not reachable, so $\delta(s,u) = \infty$.
Otherwise, sps u, v reachable.
Take any path of shortest length $\delta(s,u)$ from s to u. Then this path + (u,v) is path from s to v of length $\delta(s,u)+1$
Thus $\delta(s,v) \leq \delta(s,u)+1$ ∎

Lemma 8.2.2. *Let $G = (V, E)$ be a directed or undirected graph, and suppose that BFS is run on G from a given source vertex $s \in V$. Then upon termination, for each vertex $v \in V$, the value $v.d$ computed by BFS satisfies $v.d \geq \delta(s, v)$.*

<u>Proof</u>: (Loop invariant) Each time line 10 is run, $\forall v \in V$, $v.d \geq \delta(s,v)$.
(Initialization) First time line 10 is run, $\forall v \in V \setminus \{s\}$, $v.d = \infty \geq \delta(s,v)$ and $v.s = 0 = \delta(s,s)$.
(Maintenance) Sps line 10 just ran, $Q \neq \emptyset$ and loop invariant is true. Sps u is dequeued in line 11, and we look at $v \in G.Adj[u]$. and v is white. Then we set $v.d = u.d + 1$ but

$$\begin{aligned} v.d &= u.d+1 \\ &\geq \delta(s,u)+1 \quad \text{by loop inv.} \\ &\geq \delta(s,v) \quad \text{by prev. lemma.} \end{aligned}$$

(Termination) Since loop invariant is true at end, desired inequalities are true upon termination. $\blacksquare$

Can be at most two d-values in queue Q:

Lemma 8.2.3. *Suppose that during the execution of BFS on a graph $G = (V, E)$, the queue Q contains the vertices $\langle v_1, v_2, \ldots, v_r \rangle$, where v_1 is the head of Q and v_r is the tail. Then, $v_r.d \le v_1.d + 1$ and $v_i.d \le v_{i+1}.d$ for $i = 1, 2, \ldots, r-1$.*

Proof: We will prove this by induction on # of queue opertations.

Initially queue just contains s, so true.

(Dequeue) Sps lemma currently true and dequeue v_1. Then v_2 new head (or $Q=\emptyset$, in which case automatically true)

$v_r.d+1 \le v_1.d+1 \le v_2.d+1$ by assumption.

(Enqueue) Sps we just enqueued vertex v, so $v_{r+1}=v$. v is enqueued because of a recently dequeued vertex u. By inductive assumption, $u.d \le v_1.d$.

Thus $v_{r+1}.d = v.d = u.d+1 \le v_1.d+1$.

Inductive assumption also implies $v_r.d \le u.d+1$ so $v_r.d \le u.d+1 = v.d = v_{r+1}.d$.

All other inequalities remain true by ind. assumption. $\blacksquare$

As a consequence:

Corollary 8.2.4. *Suppose that vertices v_i and v_j are enqueued during the execution of BFS, and that v_i is enqueued before v_j. Then $v_i.d \le v_j.d$ at the time that v_j is enqueued.*

(Also uses observation that d-value never reassigned after first getting finite value)

Theorem 8.2.5 (Correctness of breadth-first search)**.** *Let $G = (V, E)$ be a directed or undirected graph, and suppose that BFS is run on G from a given single source vertex $s \in V$. Then, during its execution, BFS discovers every vertex $v \in V$ that is reachable from the source s, and upon termination, $v.d = \delta(s, v)$ for all $v \in V$. Moreover, for any vertex $v \neq s$ that is reachable from s, one of the shortest paths from s to* ~~$s.\pi$~~ $v.\pi$ *followed by the edge $(v.\pi, v)$.*

Proof: Assume towards contradiction there is v s.t. $v.d \neq \delta(s,v)$, so $v.d > \delta(s,v)$.
Let v be a counterexample w/ minimal $\delta(s,v)$.
Then v must be reachable from s. Sps u immediately precedes v on some shortest path from s. Then $\delta(s,u) < \delta(s,v)$ so

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

(†) $v.d > \delta(s,v) = \delta(s,u)+1 = u.d+1$.

consider point at which u was dequeued.

3 cases: (v white) Then $u.d+1 = v.d$ contradicting (†)

(v black) so $v.d \leq u.d$, by prev. cor contradicting (†).

(v gray) Then v was painted gray by a vertex w dequeued before u. Then $v.d = w.d+1 \leq u.d+1$, also a contradiction. ∎

Breadth-first trees

Given a graph $G=(V,E)$ w/ source s, define <u>predecessor subgraph</u> of G $G_\pi = (V_\pi, E_\pi)$ w/

$$V_\pi = \{v \in V : v.\pi \neq \text{NIL}\} \cup \{s\}$$

$$E_\pi = \{(v.\pi, v) : v \in V_\pi, v \neq s\}.$$

We say G_π is a <u>breadth-first tree</u> if V_π consists of all vertices reachable from s and for all $v \in V_\pi$, G_π contains a unique simple path from s to v that is also a shortest path from s to v in G.

Edges of E_π called <u>tree edges</u>.

Note: breadth-first tree is a tree bcs it is connected and $|E_\pi| = |V_\pi| - 1$.

Lemma 8.2.6. *When applied to a directed or undirected graph $G = (V, E)$, procedure BFS constructs π so that the predecessor subgraph $G_\pi = (V_\pi, E_\pi)$ is a breadth-first tree.*

Proof: Line 16 of BFS sets $v.\pi = u$ iff $(u,v) \in E$ and $\delta(s,v) < \infty$. So V_π consists of all vertices reachable from s. Since G_π is a tree, it contains unique simple path from s to each vertex in V_π. By prev. Thm, this is a shortest path from s to v. ∎

Assume BFS(G, s) already run.

PRINT-PATH(G, s, v)

if $v == s$
    print s
Elseif $v.\pi ==$ NIL
    print "no path from" s "to" v "exists"
else PRINT-PATH$(G, s, v.\pi)$
    print v