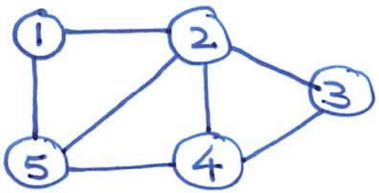
chang@umbc.edu

Graph Algorithms

Some notation:

G: 

$G = (V, E)$

V = set of vertices
$= \{1, 2, 3, 4, 5\}$

E = set of edges
$= \{(1,2), (1,5), (2,3), (2,4), (2,5), (3,4), (4,5)\}$

unordered pair

abuse of notation $(1,2) = (2,1)$

self loop $= (v, v)$ often not allowed

Size of a graph

not 2.718...

Common notation: $n = |V|$, $m = |E|$ or $e = |E|$

Book uses $V = |V|$ $E = |E|$

abuse of notation, type inferred from context

$0 \leq E \leq V^2$

So, $\Theta(E)$ running time is better than $\Theta(V^2)$

Types of graphs

Undirected: $(u,v) = (v,u)$

Directed: $(u,v) \neq (v,u)$

acyclic = no cycles

undirected graph

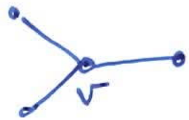
connected = for all vertices $u, v \in V$ there is a path from u to v.

a connected acyclic undirected graph = ?

an acyclic undirected graph = ?

Degrees

undirected graph: $\deg(v)$ = # of edges incident on v



$\deg(v) = 3$

directed graph:

$\mathrm{indeg}(v)$ = # of edges into v

$\mathrm{outdeg}(v)$ = # of edges out of v

$\mathrm{indeg}(v) = 3$

$\mathrm{outdeg}(v) = 2$

Degrees (cont'd)

directed graph:

$$\sum_{v \in V} \text{indeg}(v) = |E| = \sum_{v \in V} \text{outdeg}(v)$$

undirected graph:

$$\sum_{v \in V} \deg(v) = 2|E|$$

Running time proportional to degree at each vertex is $\Theta(E)$.

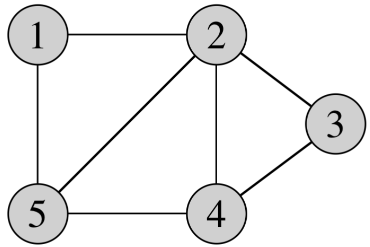
Data Structures for graphs:

Adjacency list

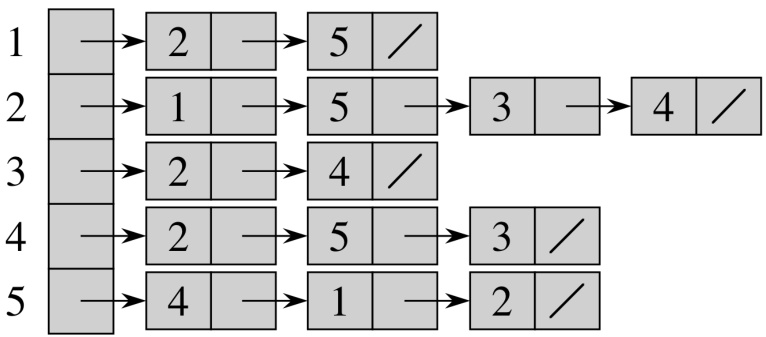
pros: running time $\Theta(V+E)$ to look at every edge
cons: more complicated

Adjacency Matrix

pros: simpler
cons: takes $\Theta(V^2)$ to look at every edge



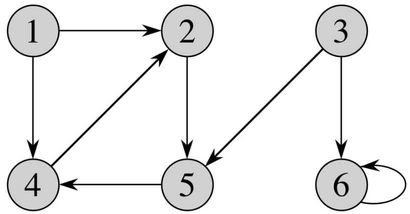
(a)



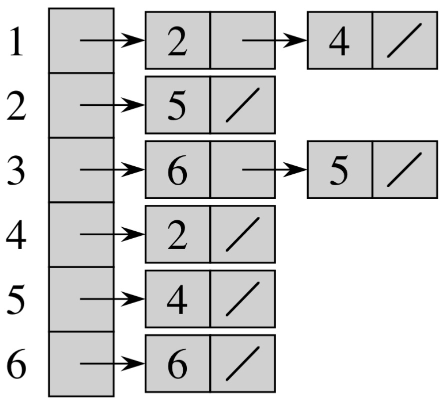
(b)

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)



(a)



(b)

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

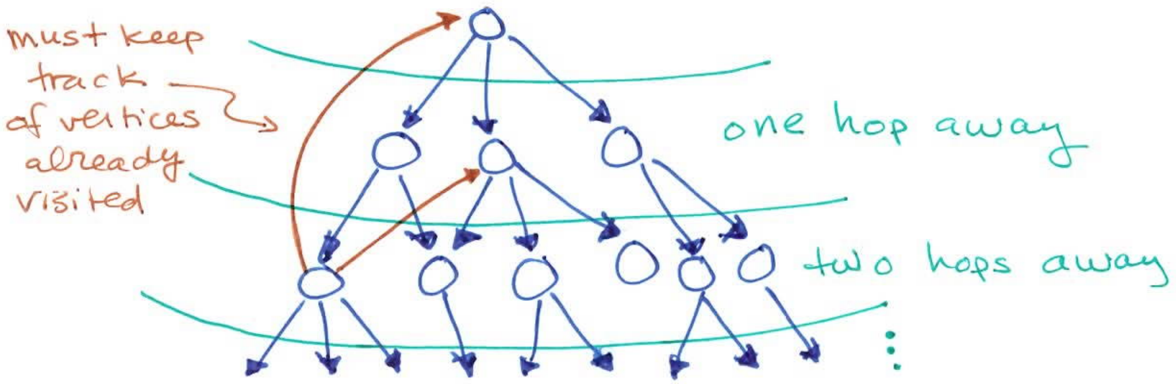
(c)

Breadth-First Search = BFS

Explore graph from start vertex s.

Is the graph connected?

# of connected components?



Classifying edges after BFS:

$\pi(v) = u$ means BFS visit from u to v.
& make (u,v) a tree edge

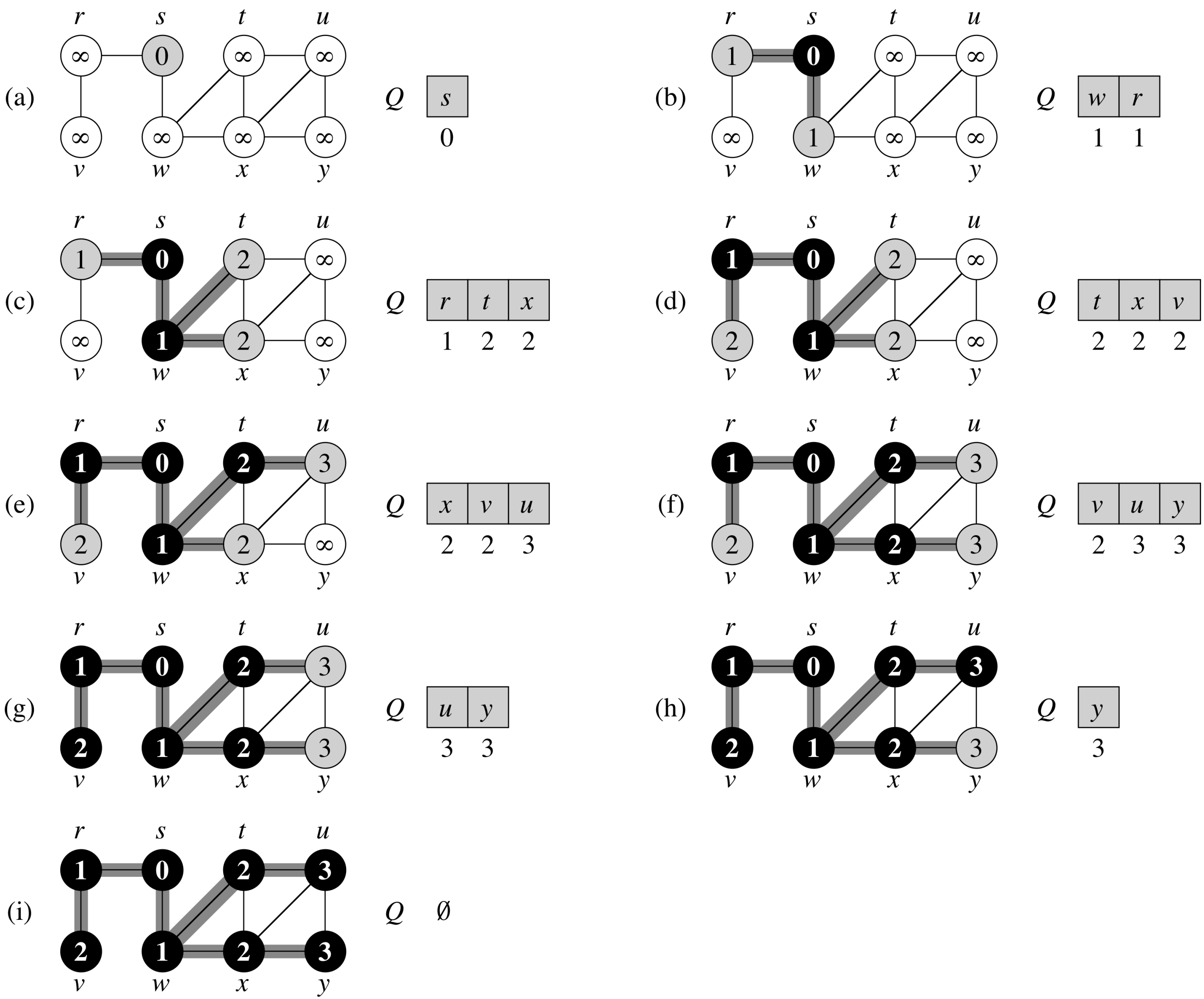
(u,v) is a back edge if v is an ancestor of u
in the BFS tree and $(u,v) \in E$

makes more sense in directed graphs

(u,v) is a forward edge if u is an ancestor of v
in the BFS tree, $(u,v) \in E$ and (u,v)
is not a tree edge.

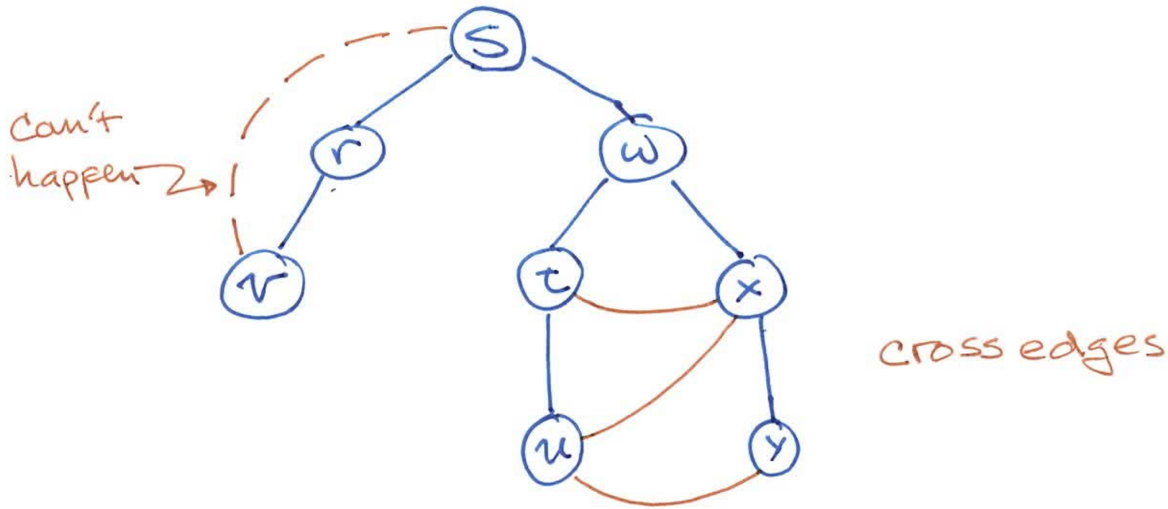
not possible

all other edges are cross edges.

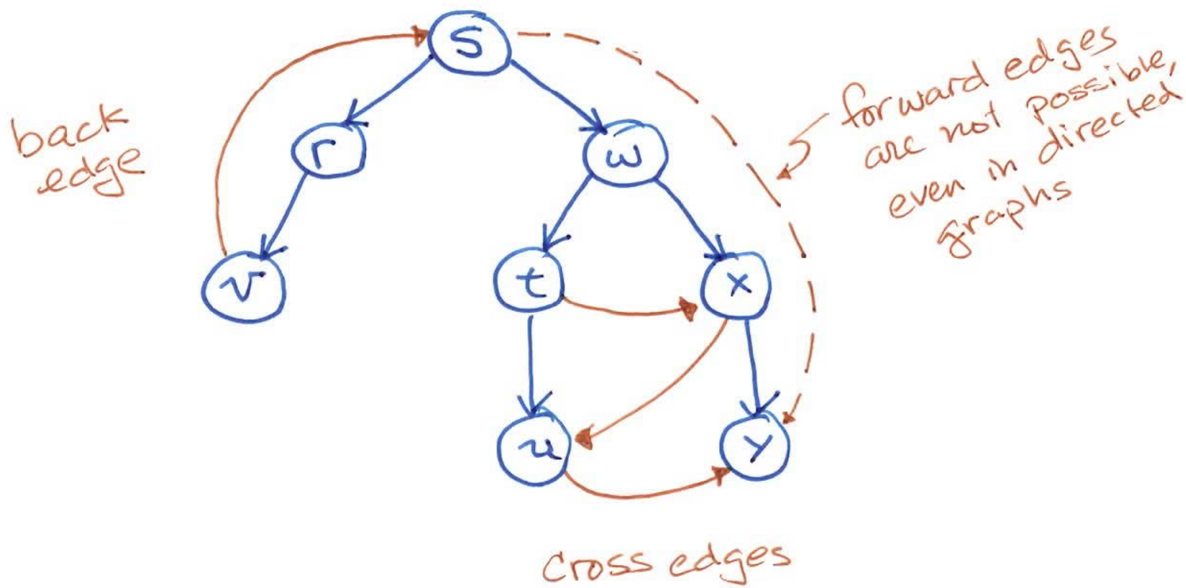


(a)
r s t u
∞ 0 ∞ ∞
∞ ∞ ∞ ∞
v w x y
Q
s
0
(b)
r s t u
1 0 ∞ ∞
∞ 1 ∞ ∞
v w x y
Q
w r
1 1
(c)
r s t u
1 0 2 ∞
∞ 1 2 ∞
v w x y
Q
r t x
1 2 2
(d)
r s t u
1 0 2 ∞
2 1 2 ∞
v w x y
Q
t x v
2 2 2
(e)
r s t u
1 0 2 3
2 1 2 ∞
v w x y
Q
x v u
2 2 3
(f)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q
v u y
2 3 3
(g)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q
u y
3 3
(h)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q
y
3
(i)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q
Ø

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



What if the graph is directed?


back edge
s
r
w
v
t
x
u
y
forward edges are not possible, even in directed graphs
cross edges

BFS observations

white = unexplored   grey = in FIFO queue

black = all done

unless it is never visited

Every vertex goes from white to grey to black.

Time to explore neighbors of v = $\deg(v)$

Time to explore graph $= \sum_{v \in V} \deg(v) + |V|$

$= \Theta(V+E)$

Prove that BFS works.

How???

BFS started here

$d[v]$ is the length of <u>some</u> path from s to v.

$\delta(s,v)$ = # of edges in <u>shortest path</u> from s to v.

Prove that after a BFS, $d[v] = \delta(s,v)$ for all $v \in V$.

Another observation:

remove from front

add to the back

$$Q = (\underbrace{v_1, v_2, v_3, \ldots}_{d[v_i]=\ell}, \underbrace{\ldots, v_r}_{d[v_i]=\ell+1})$$

cannot have $d[v_j] = \ell + 2$

Remove v_i with $d[v_i] = \ell$.

Add v_j new vertex adjacent to v_i to Q.

Then $d[v_j] = d[v_i] + 1$.

Proof outline:

Thm After BFS, $d[v_i] = \delta(s, v_i)$ for all $v_i \in V$.

Lemma 1 For all $(u, v) \in E$, $\delta(s, v) \le \delta(s, u) + 1$.

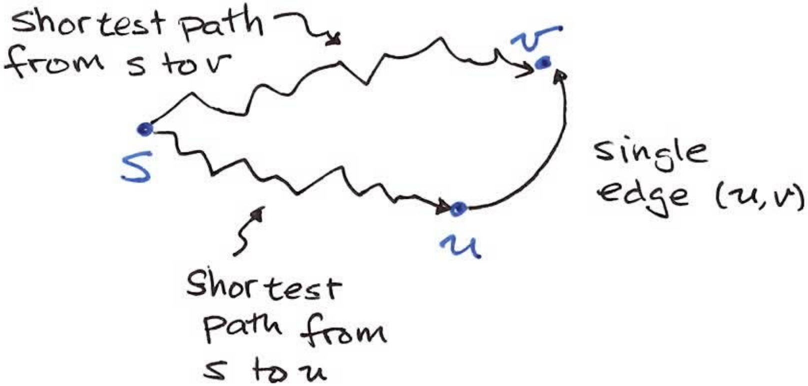
Lemma 2 For each $v \in V$, $\delta(s, v) \le d[v]$.

Lemma 3 During BFS, vertices in Q have $d[\,]$ values at most 1 apart.

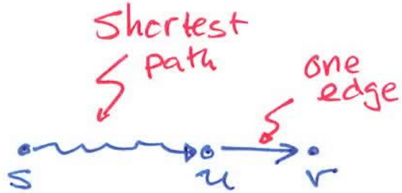
Lemma 4 If x is enqueued before y, then $d[x] \le d[y]$.

Lemma 5 For all $v \in V$, $\delta(s, v) = d[v]$.

Lemma 1 For all $(u,v) \in E$, $\delta(s,v) \le \delta(s,u) + 1$.



If $\delta(s,v) \ge \delta(s,u) + 2$, then path



has length $\delta(s,u) + 1$ and is shorter than shortest path from s to v. $\Rightarrow\!\Leftarrow$

contradiction

Lemma 2 For each $v \in V$, $\delta(s,v) \le d[v]$.

Pf: $d[v]$ is the length of path from s to v following tree edges. If v is not reachable from s, then $d[v] = \infty$ and $\delta(s,v) \le d[v]$ is true.

$$\begin{aligned}\delta(s,v) &= \text{length of shortest path from } s \text{ to } v \\ &\le \text{length of } \underline{\text{any}} \text{ path from } s \text{ to } v \\ &\le d[v]\end{aligned}$$

Lemma 3 During BFS, vertices in Q have d[ ] values at most 1 apart.

Pf: prior observation.

Lemma 4 If x is enqueued before y, then $d[x] \le d[y]$.

Pf: trivial.

Lemma 5 For all $v \in V$, $\delta(s,v) = d[v]$.

Pf: By Lemma 2, we know $\delta(s,v) \le d[v]$.

Is $\delta(s,v) < d[v]$ possible? Show "no" by contradiction.

Let $BAD = \{ x \mid \delta(s,x) < d[x] \}$. what if $BAD = \emptyset$?

Choose $v \in BAD$ with smallest $\delta(s,v)$.

Consider shortest path from s to v: $s \rightsquigarrow u \to v$

Let u be the vertex prior to v on shortest path.

$$\delta(s,v) = \delta(s,u) + 1$$

What color was v when u was dequeued?

① if v is white, then

u becomes predecessor of v.

$d[v] \leftarrow d[u] + 1$

Since $\delta(s,u) < \delta(s,v)$, $u \notin BAD$.

So $\delta(s,u) = d[u]$.

Then $d[v] = d[u] + 1 = \delta(s,u) + 1 = \delta(s,v)$

② If v is grey, then v is still in Q.

By Lemma 3, $d[v] \le d[u] + 1$.

d[ ] values in Q at most 1 apart.

Then, $d[v] \le \delta(s,v)$, since $d[u] = \delta(s,u)$

and $d[u] + 1 = \delta(s,u) + 1 = \delta(s,v)$.

③ If v is black, then v has been dequeued.

$\Rightarrow d[v] \le d[u]$ by Lemma 4.

$\Rightarrow d[v] < d[u] + 1 = \delta(s,u) + 1 = \delta(s,v)$.

$\Rightarrow d[v] < \delta(s,v)$ $\Rightarrow\Leftarrow$ contradiction