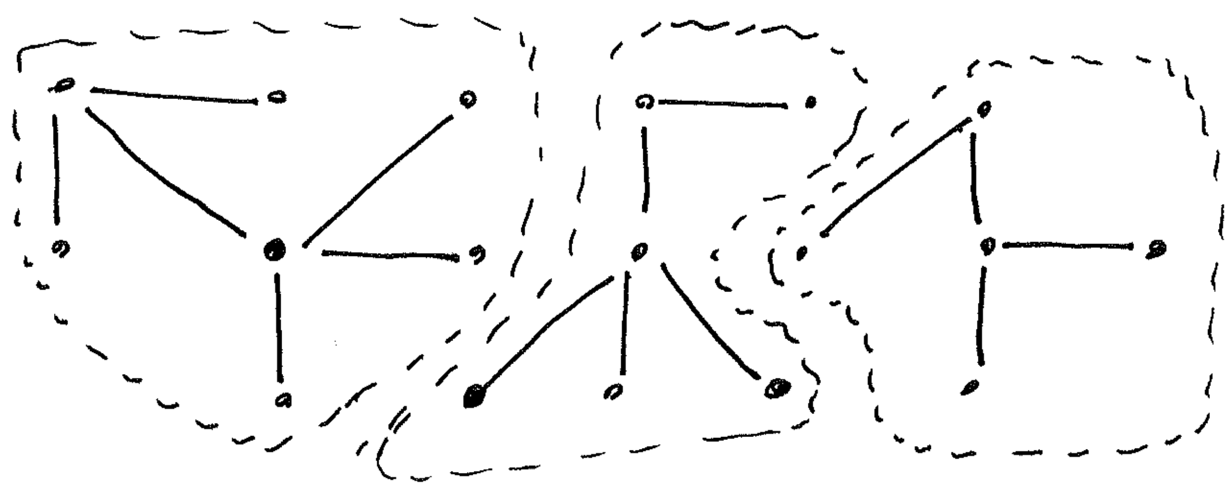
CSE 101-01 4-7-22

- Pa1 ext. 1 last day

Defn

a graph G is called acyclic (or a forest) iff G contains no cycles.

Ex.



#vert:	7	6	5
#edges:	6	5	4

Defn

A graph that is both connected and acyclic is called a tree.

Theorem

If T is a tree with n vertices and m edges, then $m = n-1$.

How do we represent Graphs (and digraphs)

- Adjacency matrix
- Incidence matrix

read handout

- Adjacency list representation:

$adj[\cdot]$ is an array of lists.

Suppose $V(G) = \{1, 2, \ldots, n\}$

Then $len(adj) = n$.

- Graph (undirected)

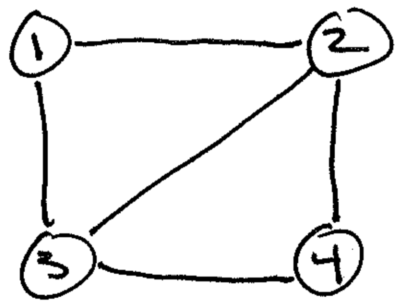
$adj[i]$: list of vertices adjacent to vertex i.

- digraph

$adj[i]$: list of terminii of edges having i as origin

Ex.

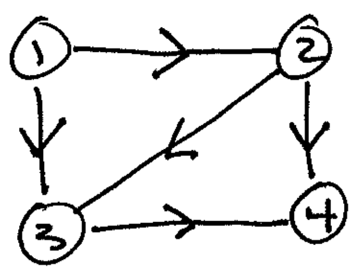
G



$$\mathrm{adj}(G) = \begin{cases} 1: & 2 \quad 3 \\ 2: & 1 \quad 3 \quad 4 \\ 3: & 1 \quad 2 \quad 4 \\ 4: & 2 \quad 3 \end{cases}$$

Ex.

G



$$\mathrm{adj}(G) = \begin{cases} 1: & 2 \quad 3 \\ 2: & 3 \quad 4 \\ 3: & 4 \\ 4: & \end{cases}$$

Handout: Graph algorithms

Defn distance function $\delta(x,y)$

$$\delta(x,y) = \begin{cases} \text{length of a shortest } x\text{-}y \text{ path if } y \text{ is reachable from } x. \\ \infty \text{ if } y \text{ is not reachable from } x. \end{cases}$$

Single source shortest paths (SSSP)

Problem:

Given a graph (or digraph) G and a vertex $s \in V(G)$ (called source)

(1) determine $\delta(s,x)$ for all $x \in V(G)$.

(2) for all $x \in V(G)$ with $\delta(s,x) < \infty$, determine a shortest s-x path

Breadth First Search (BFS)

Vertex attributes:

- color[x] : white
  gray
  black

- distance[x] : $\delta(s,x)$

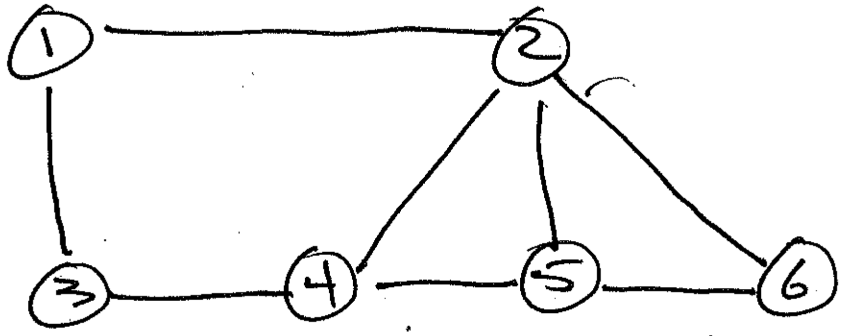
- parent[x] : predecessor of x

BFS uses a FIFO queue Q.

Pseudo-code : see /Examples/pseudocode

Ex.

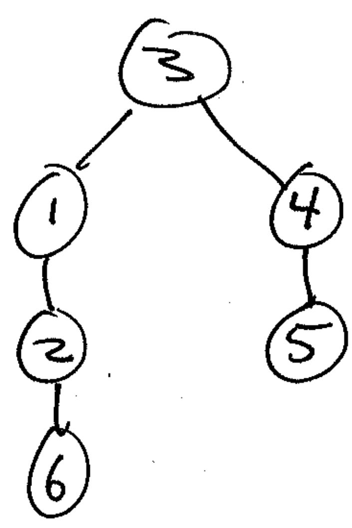
$\boxed{s = 3}$



	adj	color	d	p
1	~~2~~ ~~3~~	~~w~~ ~~g~~ b	~~∞~~ 1	~~n~~ 3
2	~~1~~ ~~4~~ ~~5~~ ~~6~~	~~w~~ ~~g~~ b	~~∞~~ 2	~~n~~ 1
3	~~1~~ ~~4~~	~~g~~ b	0	n
4	~~2~~ ~~3~~ ~~5~~	~~w~~ ~~g~~ b	~~∞~~ 1	~~n~~ 3
5	~~2~~ ~~4~~ ~~6~~	~~w~~ ~~g~~ b	~~∞~~ 2	~~n~~ 4
6	~~2~~ ~~5~~	~~w~~ ~~g~~ b	~~∞~~ 3	~~n~~ 2

Q : ~~3~~ ~~1~~ ~~4~~ ~~2~~ ~~5~~ ~~6~~

BFS Tree:



dist = depth

0

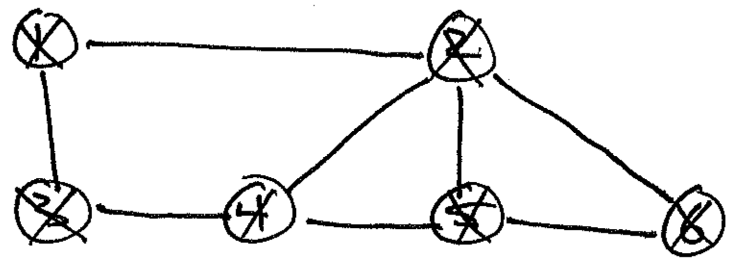
1

2

3

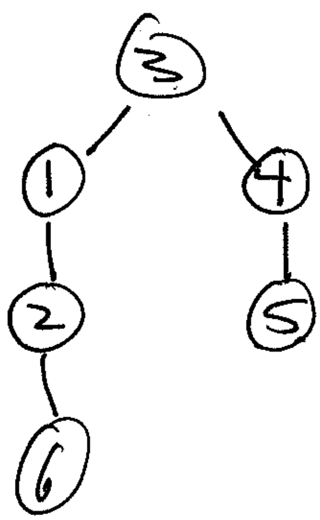
Ex Same graph, same source, Quick Way

s=3



Q: ~~3~~ ~~1~~ ~~4~~ ~~2~~ ~~5~~ 6

Tree:



dist

0

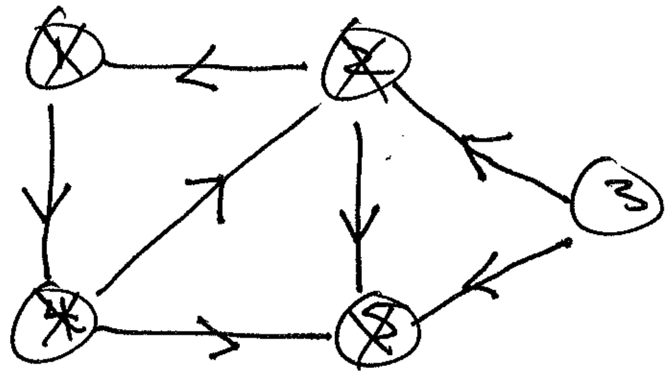
1

2

3

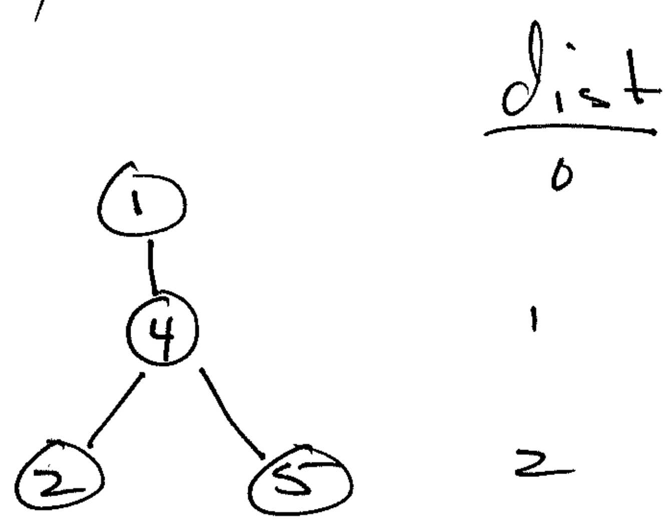
Ex.

$s = 1$



Q: $\cancel{1}$ $\cancel{4}$ $\cancel{2}$ $\cancel{5}$

Tree



	adj	color	d	p
1	4	b	0	n
2	1 5	b	2	4
3	2 5	w	∞	n
4	2 5	b	1	1
5		b	2	4

Defn

Predecessor subgraph : (V_p, E_p)

$$V_p = \{x \in V(G) \mid P[x] \neq nil\} \cup \{s\}$$

$$E_p = \{\underbrace{(P[x], x)} \mid P[x] \neq nil\}$$

ordered pair if digraph
unordered pair if graph (undir)

How to extract shortest paths from P[ ] array?

PrintPath(G, s, x)