cse 101-01 1-19-23

- Incidence matrix representation of a graph/digraph.

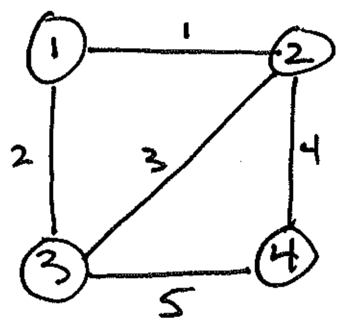
require lables:

$V = \{x_1, x_2, \ldots, x_n\}$

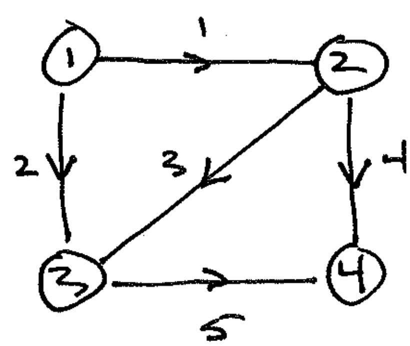
$E = \{e_1, e_2, \ldots e_m\}$

$\boxed{n \times m \text{ matrix}}$

Ex



G



D

$$I(G) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$I(D) = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

- <u>Adjacency matrix</u> representation.

requires lables

$$V = \{x_1, x_2, \ldots, x_n\}$$

$$\boxed{n \times n \text{ matrix}}$$

$$A(G) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$A(D) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- Adjacency List Representation

require labels $V = \{x_1, x_2, \ldots, x_n\}$

$\boxed{\text{Array of lists}}$

adj(G):

1 → (2, 3)

2 → (1, 3, 4)

3 → (1, 2, 4)

4 → (2, 3)

adj(D):

1 → (2, 3)

2 → (3, 4)

3 → (4)

4 → ( )

Handout Graph Algorithms

Given G (graph or digraph)
define distance from $x \in V(G)$
to $y \in V(G)$ by

$$\delta(x,y) = \begin{cases} \text{min. len. of an } x\text{-}y \text{ path} \\ \text{if } y \text{ is reachable from } x \\ \infty \quad \text{otherwise} \end{cases}$$

Single Source Shortest Paths Problem (SSSP)

given G and $s \in V(G)$ (called source vertex) determine:

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

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

Bredth First Search (BFS)

Input: adj(G) and s

Vertex attributes: $x \in V(G)$

- color[x] : w : undiscovered
  g : discovered, not finished
  b : finished

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

- Parent[x] : the predecessor of x along a shortest s-x Path.

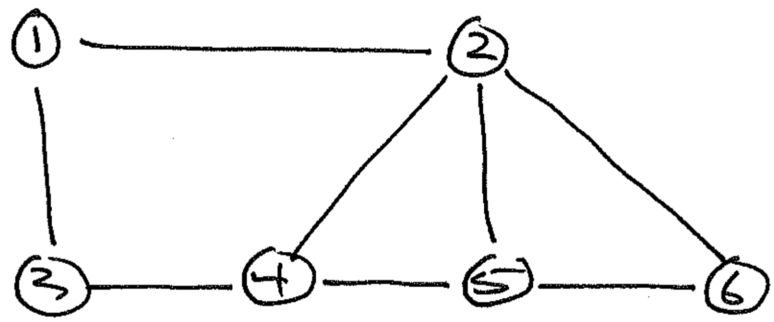
- adj[x] : adjacency list of x

Also: A fifo queue Q

see Examples / Pseudo-code

Ex. source $\boxed{s = 3}$

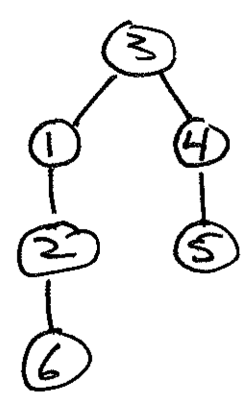
G



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

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

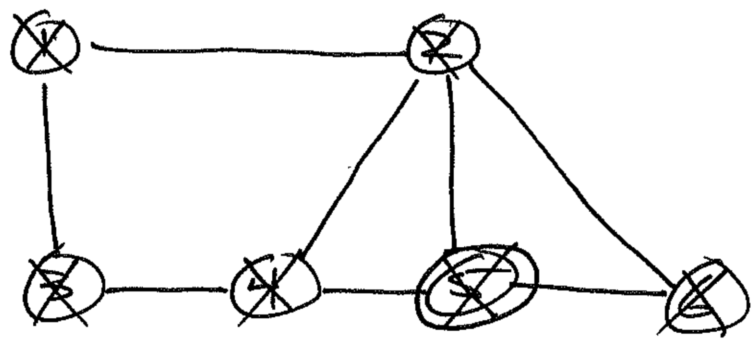
BFS Tree!



dist
0
1
2
3

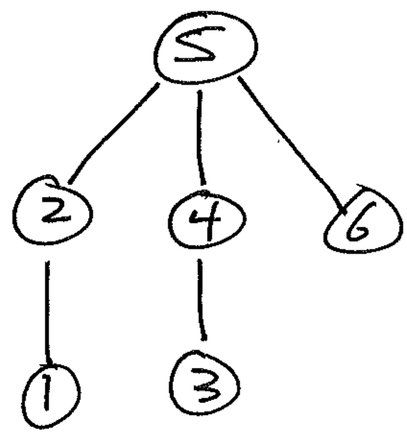
Ex. same graph, s = 5

G



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

BFS Tree:



dist

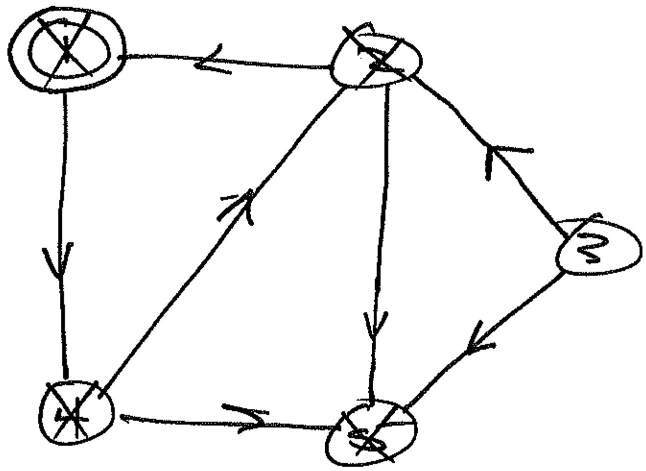
0

1

2

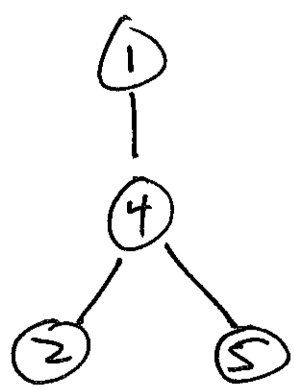
Ex. $\boxed{s = 1}$

G



Q: ~~1~~ ~~4~~ ~~2~~ ~~5~~

Tree



dist

0

1

2