CSE 101 1-18-24

- Paz Due next Fri.
- Pal ext. 1 day to sat 1-20-24.

Representations of Graphs

Let $|V| = n$, $|E| = m$

Incidence Matrix

require

$$V = \{1, 2, \ldots, n\}$$

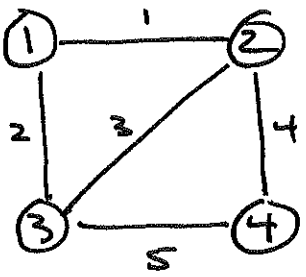
$$E = \{e_1, e_2, e_3, \ldots, e_m\}$$

$I(G)$ is an $n \times m$ matrix

- undirected:

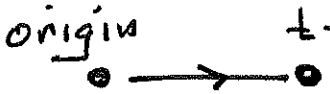
$$I_{ij} = \begin{cases} 1 & \text{if } i \text{ incident with } e_j \\ 0 & \text{otherwise} \end{cases}$$

Ex.

G 

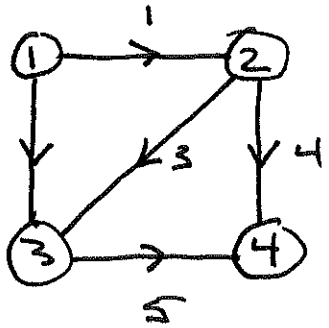
$$I(G) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

- Directed



$$I_{ij} = \begin{cases} 1 & \text{if } i \text{ is terminus of } e_j \\ -1 & \text{if } i \text{ is origin of } e_j \\ 0 & \text{otherwise} \end{cases}$$

Ex.

D 

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

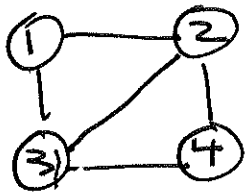
Adjacency Matrix : $A(G)$ $n \times n$ matrix

require: $V = \{1, 2, \ldots, n\}$

- undirected

$$A_{ij} = \begin{cases} 1 & \text{if } i \text{ adjacent to } j \\ 0 & \text{otherwise} \end{cases}$$

Ex. same

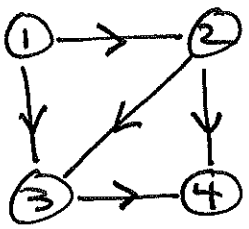


$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

- Directed

$$A_{ij} = \begin{cases} 1 & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$$

Ex



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

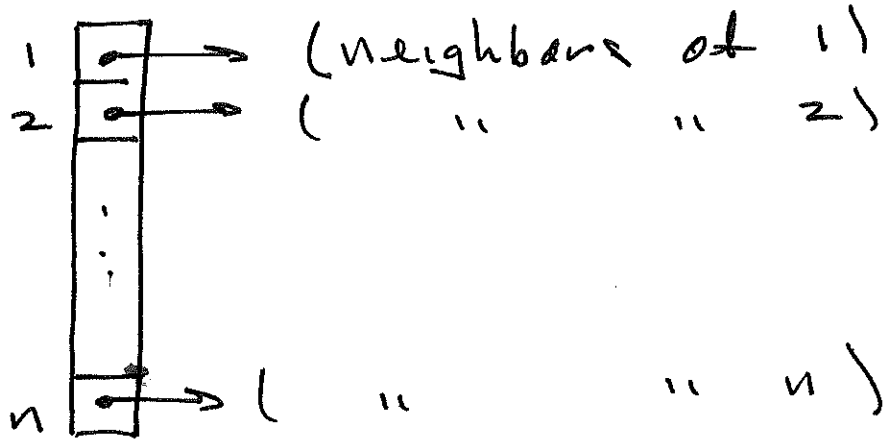
Adjacency List

1-dimensional array of lists, length of array is $n = |V|$.

require: $V = \{1, 2, \ldots, n\}$.

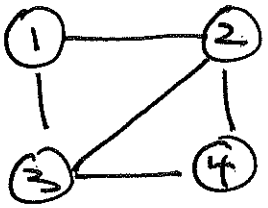
- undirected:

adj[ ]



Ex.

G



adj(G)

1 : 2 3

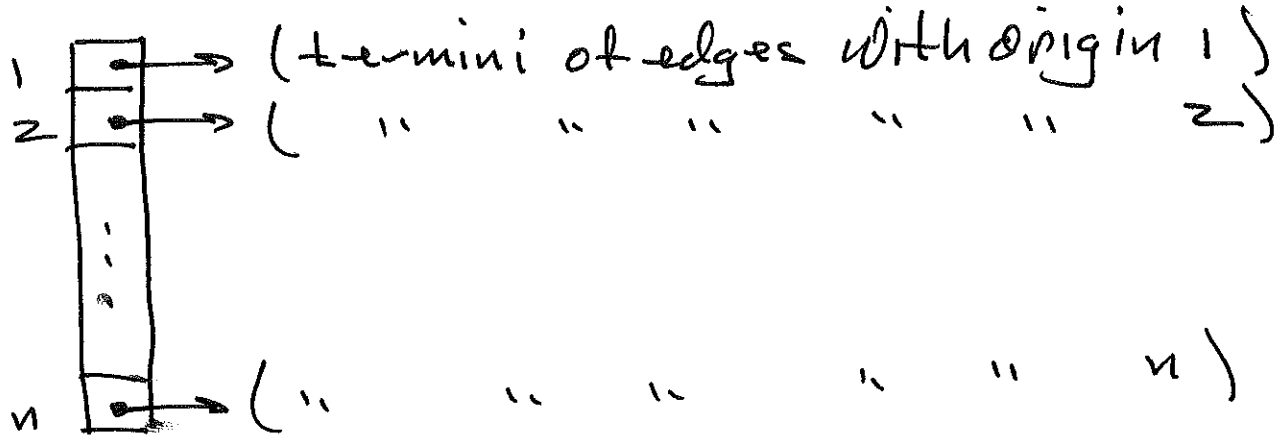
2 : 1 3 4

3 : 1 2 4

4 : 2 3

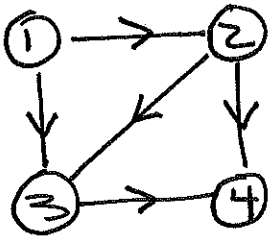
Directed

adj[ ]



Ex.

D



adj(D)

1	:	2	3
2	:	3	4
3	:	4	
4	:		

Handout: Graph Algorithms

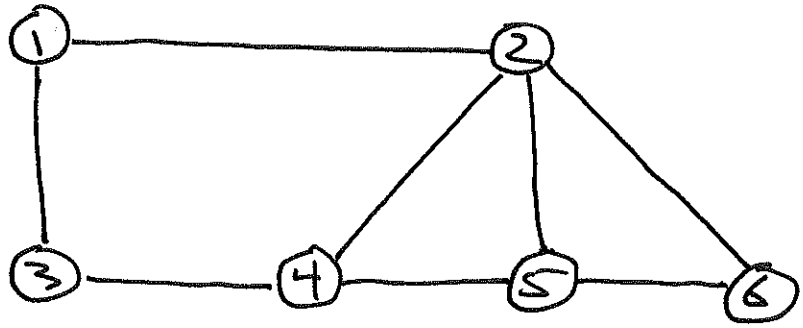
Given G (dir. or undir.) and $x, y \in V(G)$, the distance $\delta(x,y)$ from x to y is

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

Problem: single source shortest paths (SSSP)

Given G (dir. or undir.) and $s \in V(G)$ (called source): (1) determine $\delta(s,x)$ for all $x \in V(G)$, and (2) for each x with $\delta(s,x) < \infty$, determine a shortest s-x path.

Ex



$\delta(1,6) = 2$

all 1-6 Paths

- 1, 2, 6 len = 2 ← shortest
- 1, 3, 4, 2, 6 len = 4
- 1, 2, 4, 5, 6 4
- 1, 2, 5, 6 3
- 1, 3, 4, 5, 6 4

⋮ ⋮

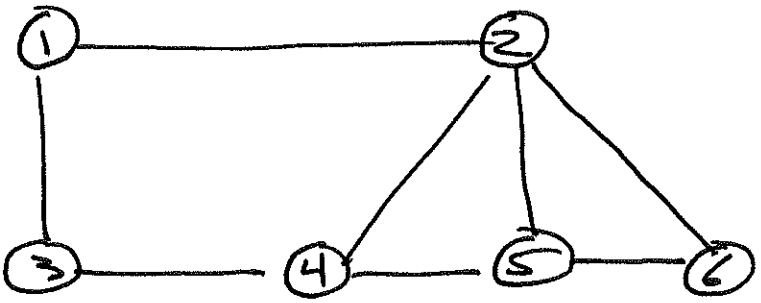
Breadth First Search (BFS)

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

- color[x] : white, gray, black
- distance[x] : distance estimate, $\delta(s,x)$ when done
- Parent[x] : Predecessor of x in a shortest s-x path.
- adj[x] : list of neighbors of x.

Will use a FIFO Queue : Q.

Ex

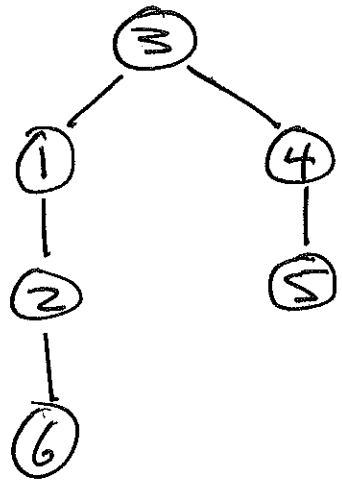


s=3

	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!



depth

0

1

2

3