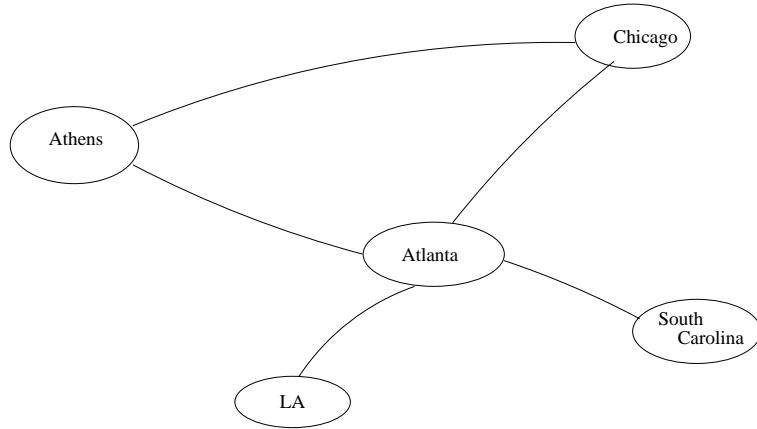

2 GRAPHS

Consists of a set of nodes or point some of which are connected by edges or lines.



A (hypothetical) graph of nonstop airline flights.

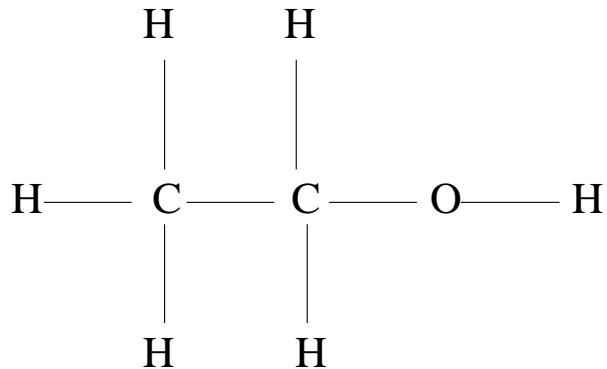
Molecule alcohol: CH_3CH_2OH

Def. A graph $G = (V, E)$ which V is the set of nodes,

$$V = \{v_1, v_2, \dots, v_n\}$$

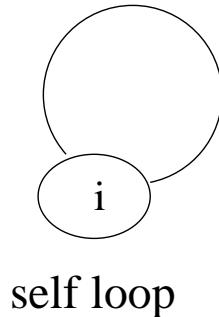
and E is the set of edges,

$$E = \{\{v_i, v_j\} | v_i \in V, v_j \in V\}$$

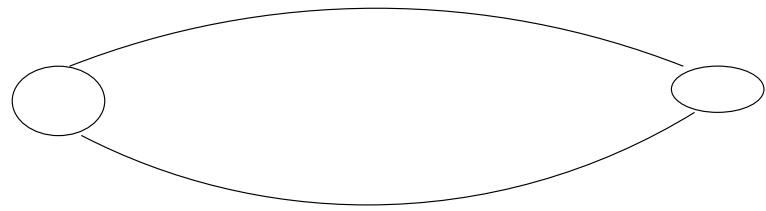


note:

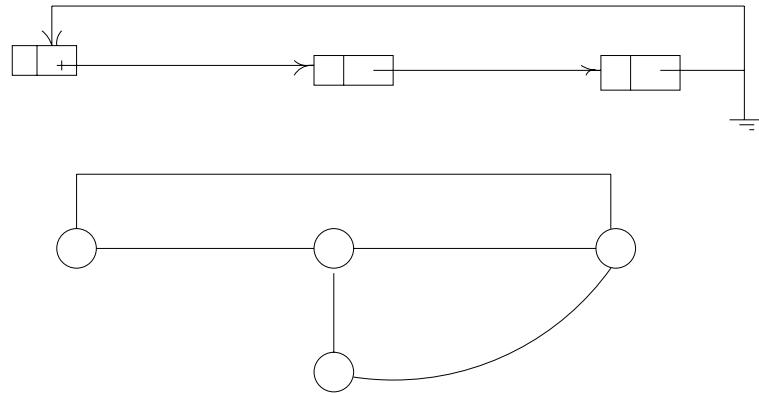
1. each edge is a set of pair of vertexes. No order implied.
2. $v\{v_i, v_j\}$, i could be equal to j .
self-loop



3. One also allows multiple edge between two node in a general graph
4. Cardinality of V , $|V| = n$, number of nodes, $|E| = m$, number of edges.



3 DIGRAPH



eg. A flow chart of a program

def. A graph whose edges are directed is a digraph, directed graph.

def. A digraph $G_2(V, E)$ when V is the set of vertex $V = \{v_1, v_2, \dots, v_n\}$ and E is the set of directed edges on every $E = \{(v_i, v_j) \text{ such that } v_i, v_j \in V\}$

Note

For the purpose of this chapter, we assume that V , the set of nodes, is nonempty & finite and that there no self loops or multiple edges in a graph or digraph.

Question

1. which route is cheapest?
2. which route is fastest?
3. if a node or a computer goes down in a computer network, does it get disconnected?
4. is there a loop in a flowchart?

3.1 Subgraph

$$V' \subseteq V, E' \subseteq E$$

Induced Subgraph by V'

$$G' = (V', E'), E' = \{u, v | u, v \in V'\}$$

Complete graph

$E = \{\{v_i, v_j | 1 \leq j \leq n\}\}$ eg. if (v, w) is an edge then v & w are adjacent to each other and v and w are said to be incident with the edge (v, w) .

def A path from v to w is a sequences of vertex v_0, V_1, v_2, \dots , such that $v_0 = v, v_k = w \& \{V_i, v_j\} \in E$.

(Books defenition is not good)

If $v_0 = v, v_k = w \&$ all v_0, v_1, \dots, v_k are distinct, then the length of the path is k . v_0 is a path of length 0.

Connected Graph:

Cycle is path $v_0v_1v_2 \dots v_k$ such that $v_0 = v_k$
A graph without any cycle is called acyclic.

Tree: Acyclic connected graph

Rooted tree has a designated root vertex establishing parent & child relationship

number of edge in a tree is $n - 1$

Could be proved by induction

A connected component of a graph G is a maximal connected subgraph of G

Weighted Graph

$G = (V, E, w)$

$W : E \leftarrow 2+$

$w(e)$ weight of e (Capacity or distance)

3.2 Representation of a graph

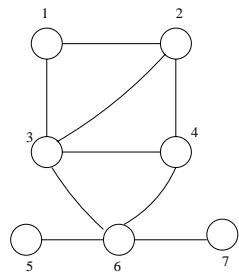
3.2.1 Adjacency Matrix

$$A = (a_{ij})_{n \times n}$$

$a_{ij} = 1$ if $v_i, v_j \in E$

0 else

for $1 \leq i, j \leq n$

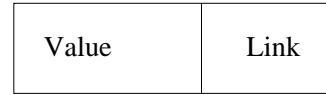
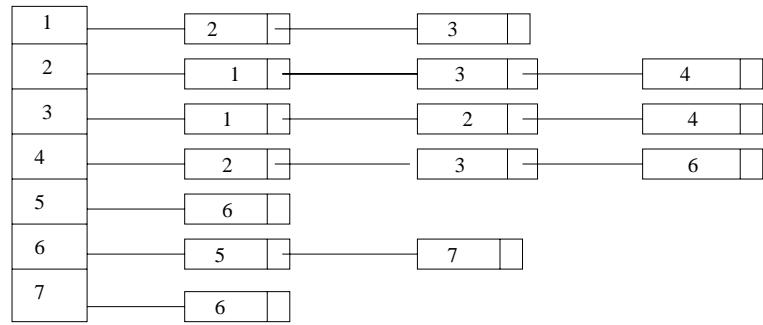


	1	2	3	4	5	6	7
1	0	1	0	1	0	0	0
2	1	0	1	1	0	0	0
3	1	1	0	1	0	0	0
4	0	1	1	0	0	1	0
5	0	0	0	0	0	1	0
6	0	0	1	1	0	1	1
7	0	0	0	0	0	1	0

Space usage $\Leftarrow O(n^2)$

3.2.2 Adjacency list

On any $A[1 \dots n]$ of linked list, $A[j]$ is the linked list of all the vertices adjacent to v_j .



Space usage $\Leftarrow O(n) + O(m) = O(m + n)$
 $= O(n^2)$ if m is $O(n^2)$

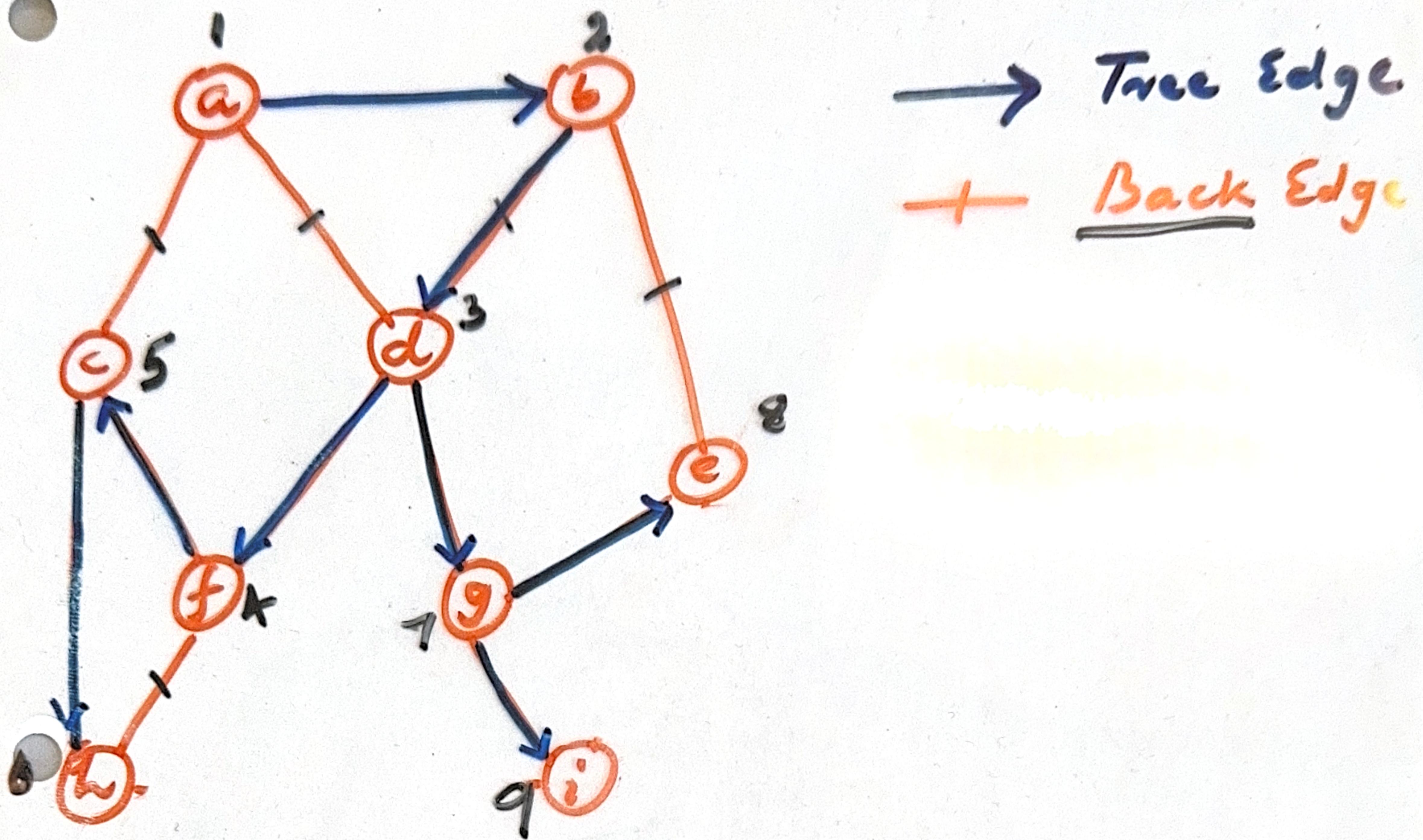
3.3 Weighted Graph Representation

Adjacency Matrix $a_{ij} = W(v_i, v_j)$ if $\{v_i v_j\} \in E$

= 0 otherwise

Adjacent link

Traversing Graphs



$DFS(v)$: Depth-First Search

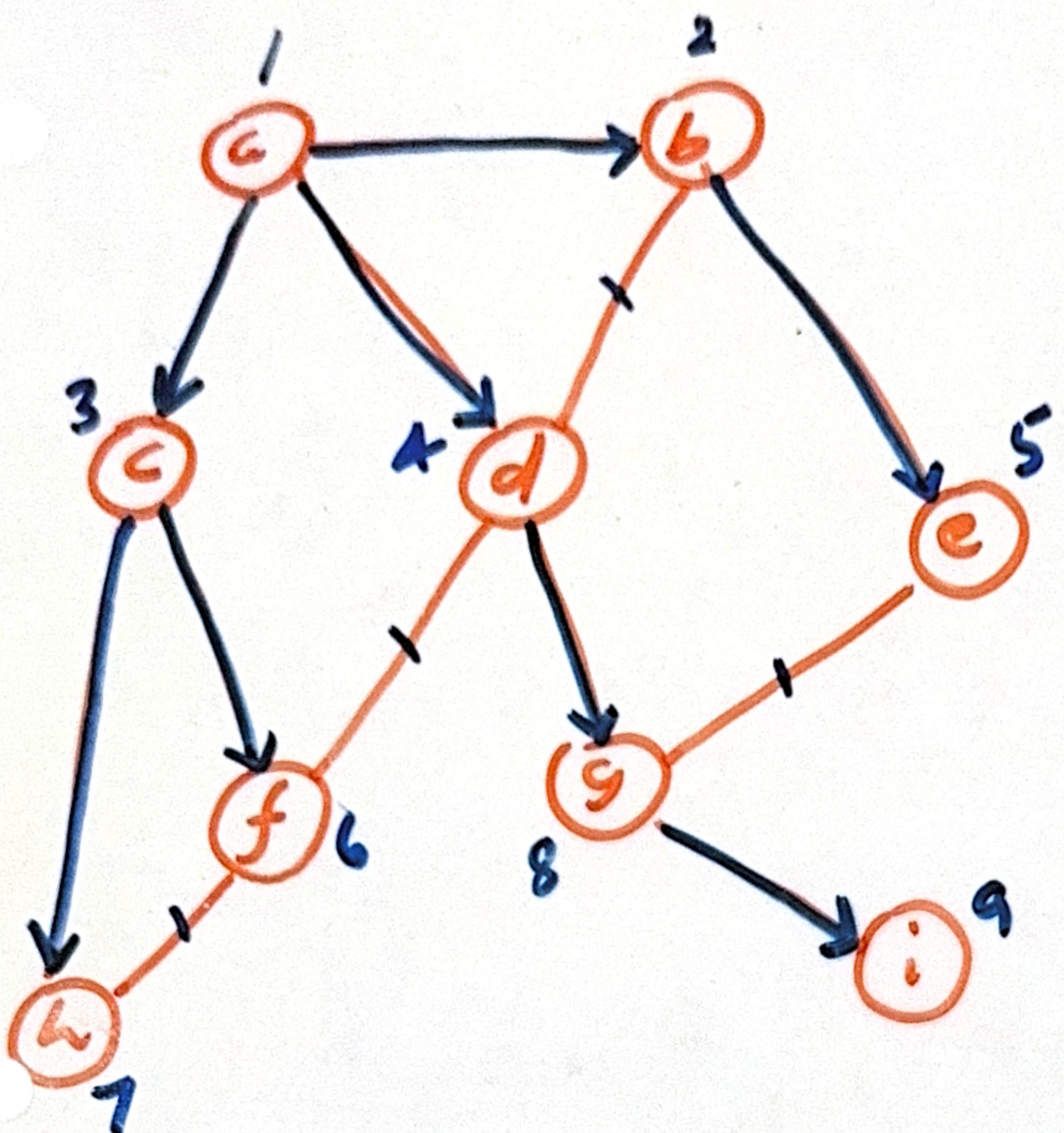
mark v

while there is an unmarked node
 w adjacent to node v

$DFS(w)$

end

Connected Components!



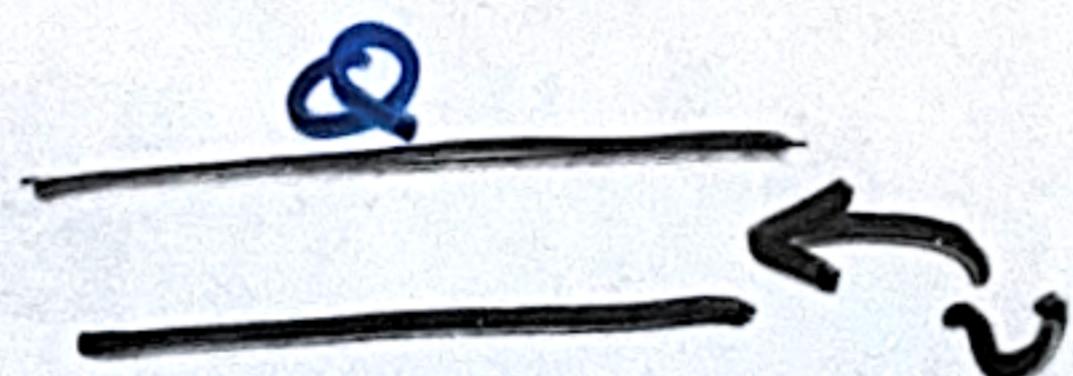
$O(m+n)$

+
cross edges

BFS (v) : Breadth-First Search

mark v

$Q = Qv$



while Q is nonempty do

$x = \text{deleteFront}(Q)$ \xleftarrow{x}

for each unmarked node
 w adjacent to x do
 { mark w
 $Q = Qw$

end

4 TRAVERSING GRAPHS

DFS(v): Depth-First Search

mark v

while there is an unmarked node w adjacent to node v DFS(w)

end

connected components

$O(m + n)$

BFS(v): Breadth-First Search

mark v

$Q = Q_v$

while Q is nonempty do

$x = \text{delete Front}(Q)$

for each unmarked node w adjacent to x do mark w

$Q = Q_w$