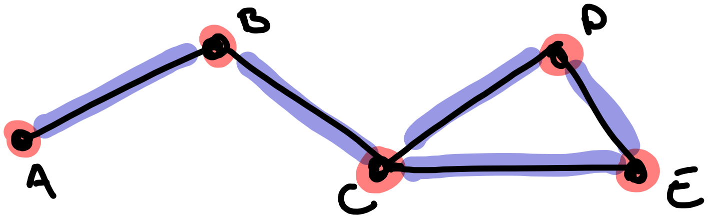
CS 1800: day16

Admin:
- HW5 due Friday
- HW6 released Friday

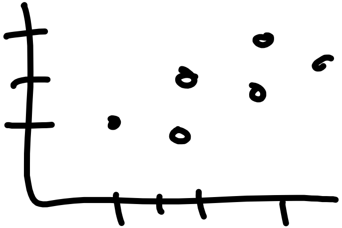
Content:
- graph definitions & anatomy
- graph representation
    - list of lists
    - adjacency matrix
- graph equivilence (isomorphism)

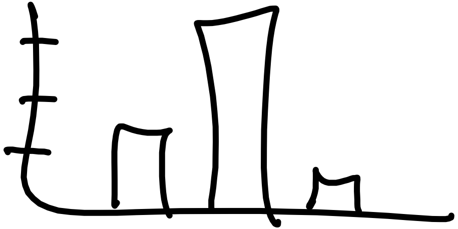
Whats a graph?

A set of nodes (vertex) and a set of edges (and edge is a pair of nodes)



More commonly, folks use the word "graph" to mean figure (as below). This is a different kind of graph. Many tech types use the word "figure" to describe these, no universal convention



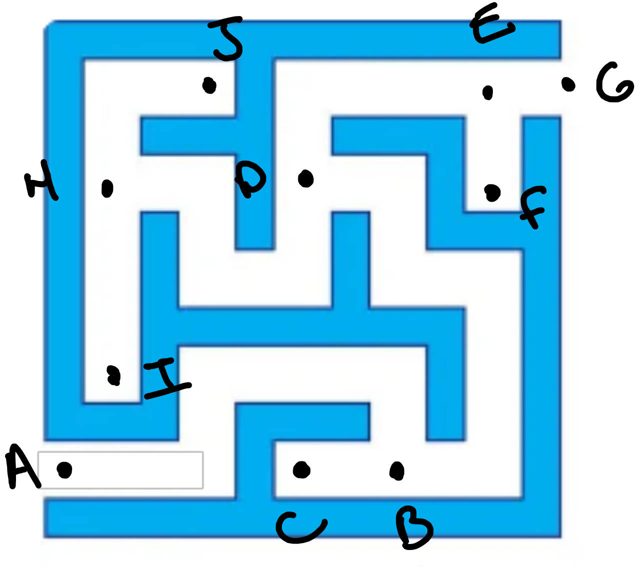


A DIFFERENT KIND OF GRAPH

Graph: Whats it good for?

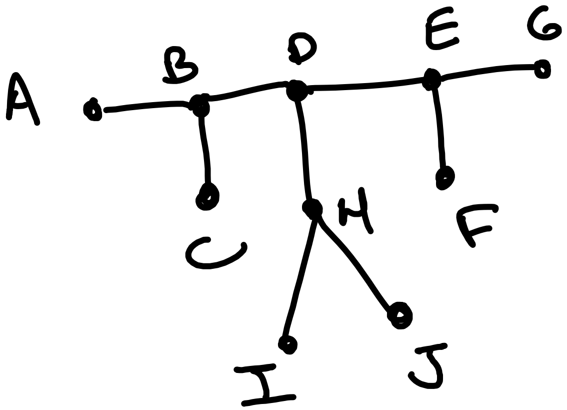
Graphs are wonderful for representing things. Often, representing clearly is a big help!

Example: represent a maze as a graph.



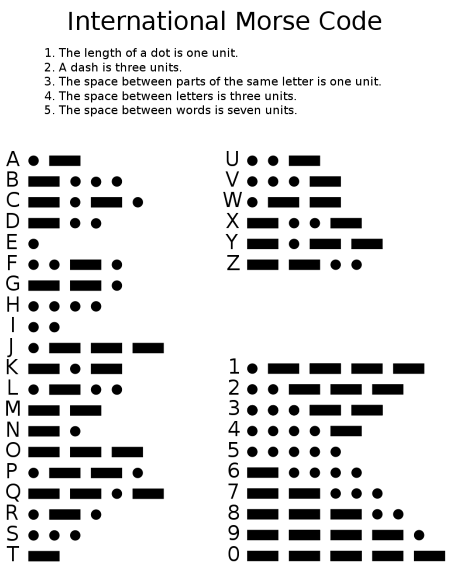
Node = intersection in maze
(start / end / dead-end too)

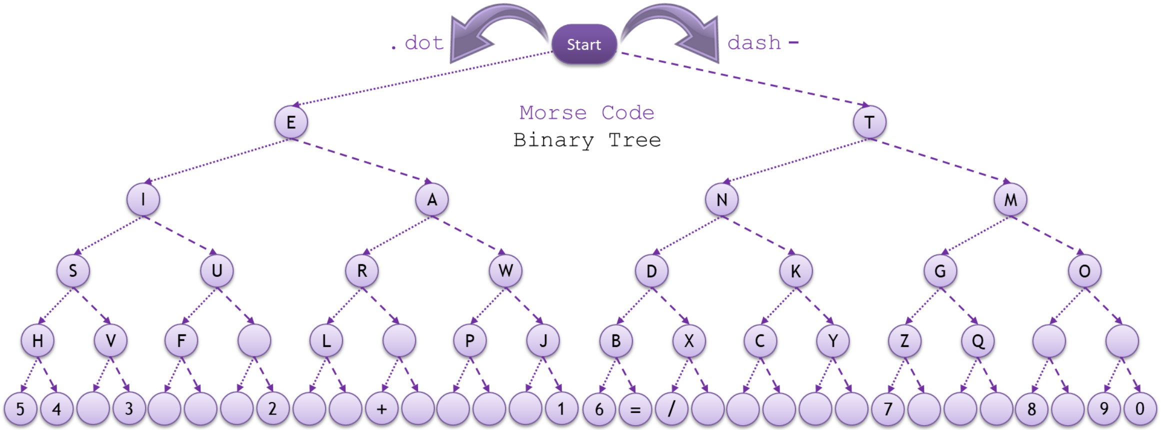
Edge = possible movement between
intersections



Graph: Whats it good for?

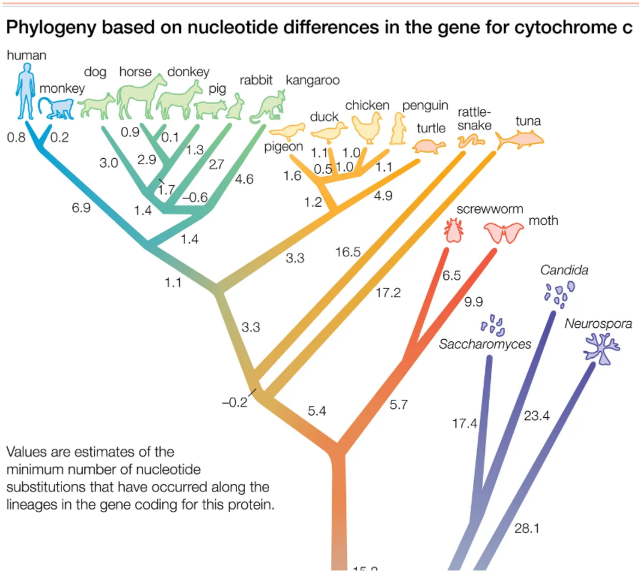
Graphs are wonderful for representing things. Often, representing clearly is a big help!





Graph: Whats it good for?

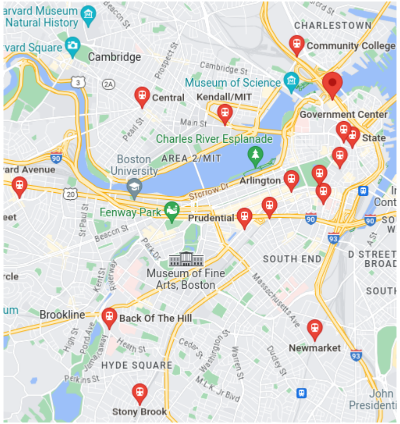
Graphs are wonderful for representing things. Often, representing clearly is a big help!



https://cdn.britannica.com/03/403-050-F1B9349F/Phylogeny-differences-cytochrome-c-protein-sequence-organisms.jpg

Graph: Whats it good for?

Graphs are wonderful for representing things. Often, representing clearly is a big help!



Whats a graph?

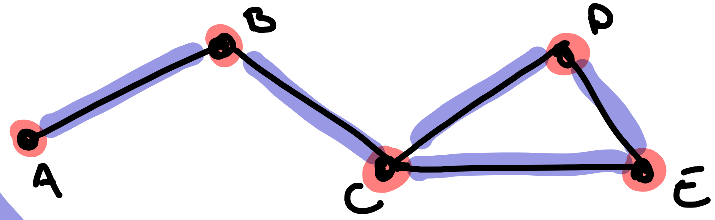
A set of nodes (also known as: vertex / vertices)

$$V = \{A, B, C, D, E\}$$

A set of edges (each edge is a pair of nodes)

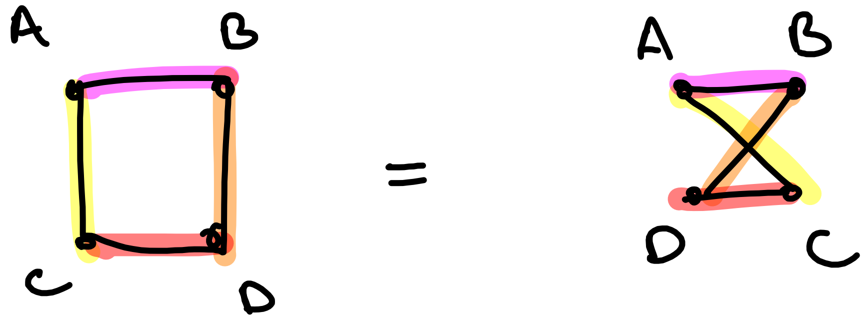
$$E = \{\{A,B\}, \{B,C\}, \{C,D\}, \{D,E\}, \{C,E\}\}$$

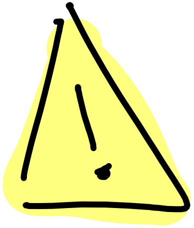
GRAPH REPRESENTED BY V, E TO LEFT:



GRAPHS CAN BE DRAWN DIFFENTLY

BUT ITS STILL SAME GRAPH



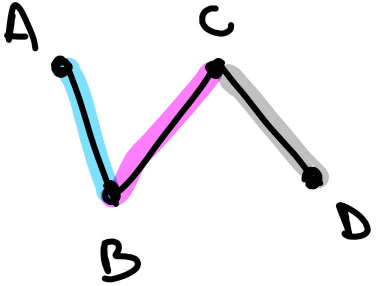


Warning:

There are a lot of terms referring to graph "stuff"

Most are super intuitive, but please double check definitions

Graph: Adjacency (undirected)



Two nodes are adjacent in a graph if there is an edge between them.

ADJACENT: A, B

NOT ADJACENT: A, C

A node and an edge are adjacent if the node is in the edge (remember, edge = pair of nodes)

ADJACENT: A, {A, B}

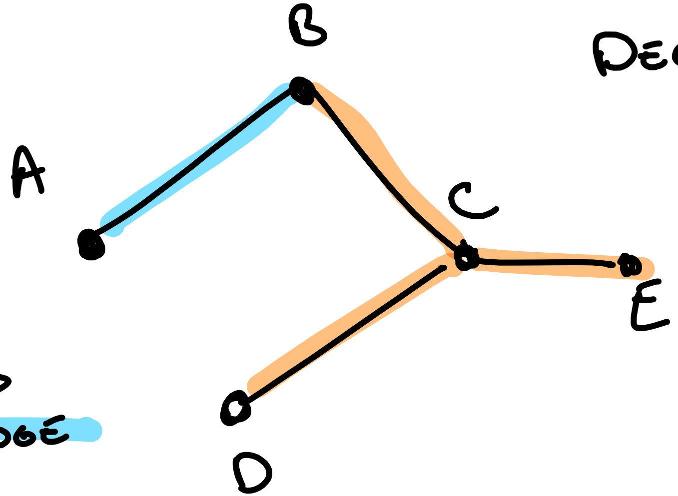
NOT ADJACENT: C, {A, B}

Two edges are adjacent if one node is adjacent to both

ADJACENT: {A, B} {B, C}

NOT ADJACENT: {A, B} {C, D}

A node's degree is the number of edges which are adjacent to it

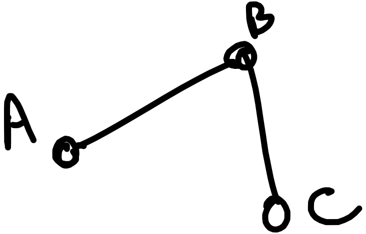


DEG(A) = 1
SINCE A HAS
1 ADJACENT EDGE

DEG(C) = 3 SINCE C HAS
3 ADJACENT EDGES

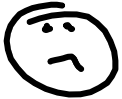
In Class Activity:

Draw a graph where the sum of degrees of all nodes is odd (or argue why this isn't possible)



$$\text{DEG}(A) + \text{DEG}(B) + \text{DEG}(C)$$
$$1 + 2 + 1 = 4$$

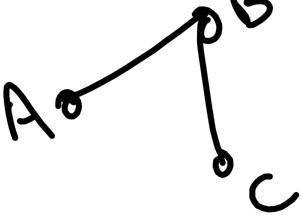
NOT ODD



What is the relationship between the following values:

- the sum of degrees for all nodes
- the number of edges in the graph

Stuck? Draw some little examples until you have your own eureka moment (really, its fun!)



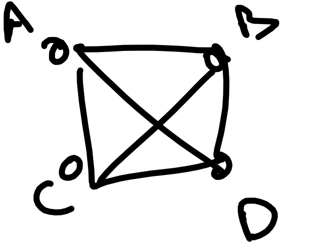
2 EDGES

$1+2+1=4$

↑ SUM OF DEGREES

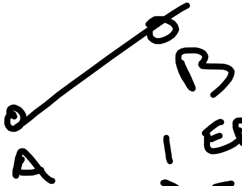
0 EDGES

$0 = \sum_i \text{DEG}$



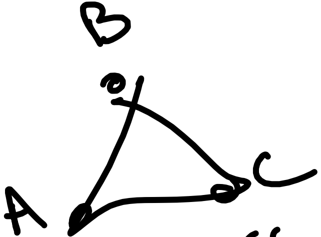
6 EDGES

$12 = \sum \text{DEG}$



1 EDGE

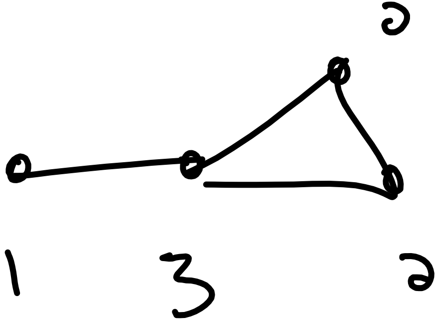
$2 = \sum \text{DEG}$



3 EDGES

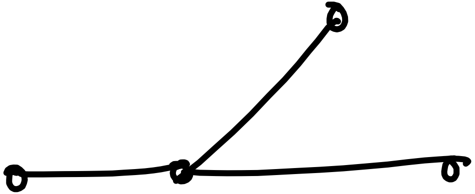
$6 = \sum \text{DEG}$

$\sum$ DEG OF ALL NODES $= 2 \cdot \#$ EDGES



4 EDGES

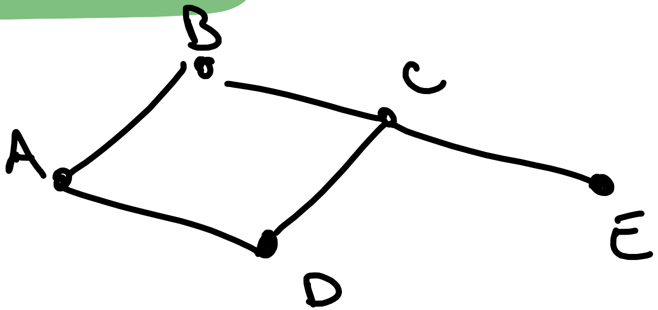
$8 = \Sigma$ DEG



Σ DEG = 6

3 EDGES

Adjacent Sequences on a Graph (walk / path / cycle):



Walk - a sequence of adjacent edges. (equivilently: a sequence of adjacent nodes)

$(\{A,B\}, \{B,C\}, \{C,B\}) = ABCB$

NOT A PATH

Path - a walk where each node is unique

ABCE

NOT A CYCLE

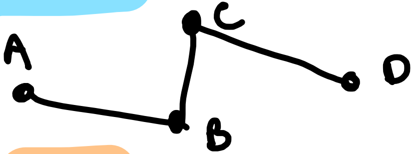
Cycle - a path which starts and ends at the same node (only this last node is not-unique)

ABCDA

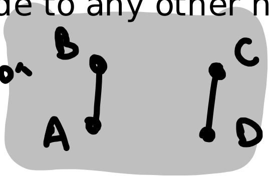
Trees! (a super useful construction)

A graph is connected if there exists a path from every node to any other node

CONNECTED:

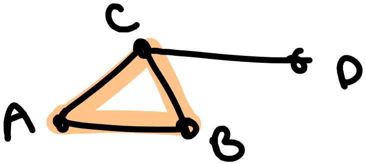


NOT CONNECTED:



GRAPH WITH 4 NODES

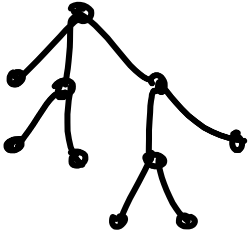
Reminder: A cycle is a path (sequence of unique, adjacent edges) which starts and ends at the same node



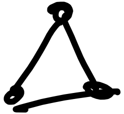
CYCLE = A B C A

Tree - a connected graph without any cycles

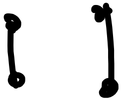
TREE:



NOT A TREE (HAS CYCLE)



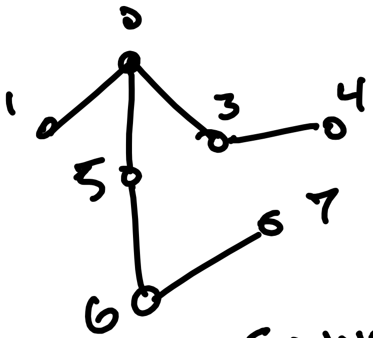
NOT A TREE: DISCONNECTED



In Class Activity

Identify a relationship between:
- the total edges in a tree
- the total nodes in a tree

Remember: a tree is connected and doesn't contain any cycles



CONNECTED ✓

NO CYCLES ✓

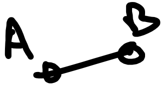
7 NODES

6 EDGES

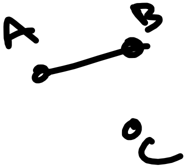
approach:
- draw some little examples
- make a conjecture (a guess as to the relationship)
- argue with your conjecture
- if you believe it, explain why your conjecture is true

TREE

CONNECTED

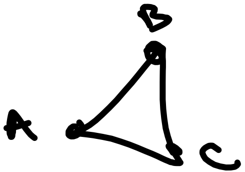


CONNECTED

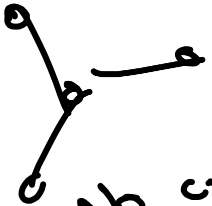


NOT

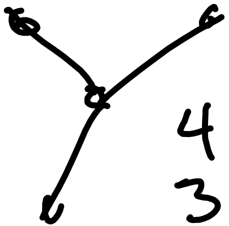
NO CYCLES



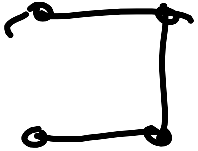
CYCLE EXISTS



NO CYCLES

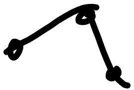


4 NODES
3 EDGES



4 NODES
3 EDGES

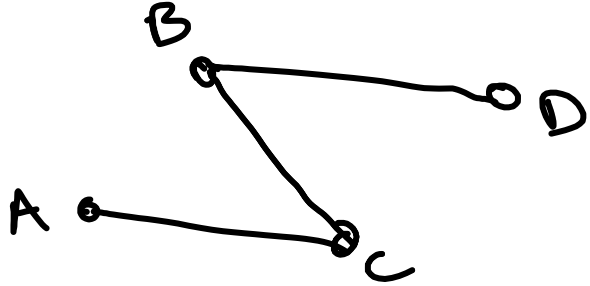
2 NODES
1 EDGE



3 NODES
2 EDGES

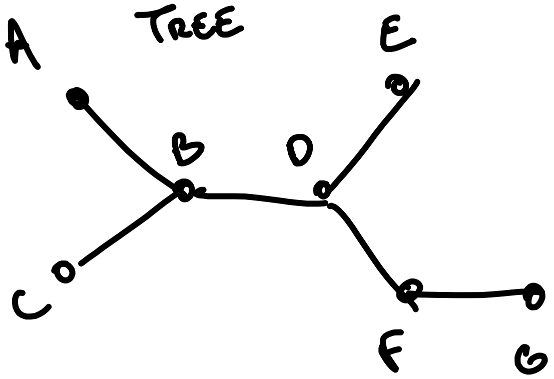
IN A TREE

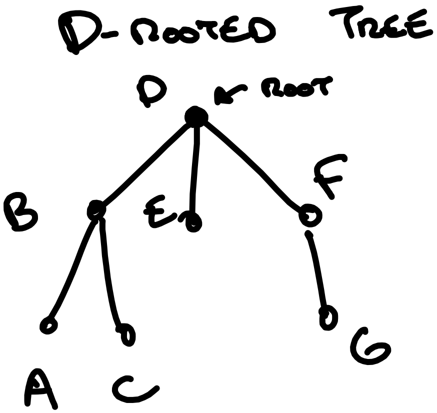
NODES-1 = # EDGES

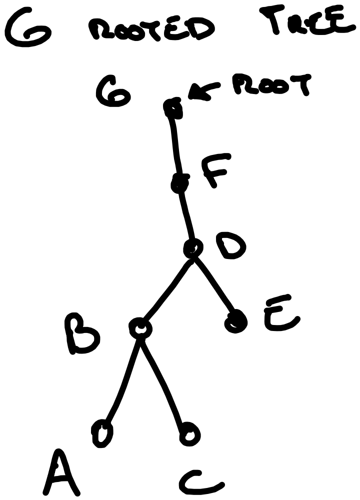


Rooted Trees

Rooted Tree - a tree (connected, acyclic graph) which has one special node identified as the root







CONVENTION

ROOT OF TREE ON TOP

(NOT RECCOMENDED FOR CHRISTMAS)

(useful fact about trees: there is a UNIQUE path between every pair of nodes)

Rooted Trees: Why go through the trouble?

Allows us to define family relationships:

parent of a node x: next node on path from x to root (root has no parent)
    ex: D is the parent of B

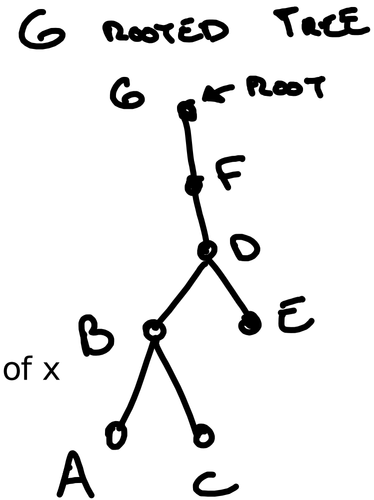
children of node x: the set of all nodes which x is parent of
    ex: {B, E} are children of D

a node is a leaf if it has no children:
    ex: A, C and E are leafs

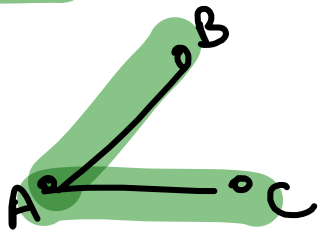
sibling of node x: the set of all nodes which whose parent is also the parent of x

ancestor of x: all nodes on the path to root

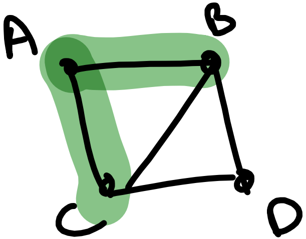
descendant of x: all nodes which have x as an ancestor



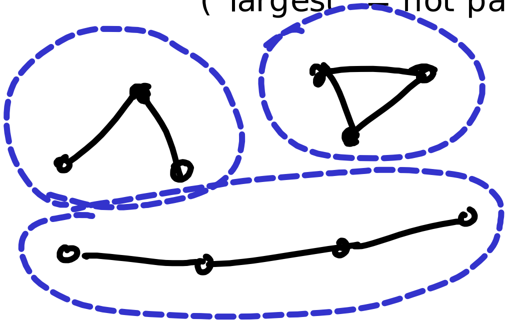
A subgraph is a graph whose nodes & edges are contained with another graph



IS SUBGRAPH OF



Connected Component- a "largest" connected subgraph
("largest" = not part of any larger connected subgraph)

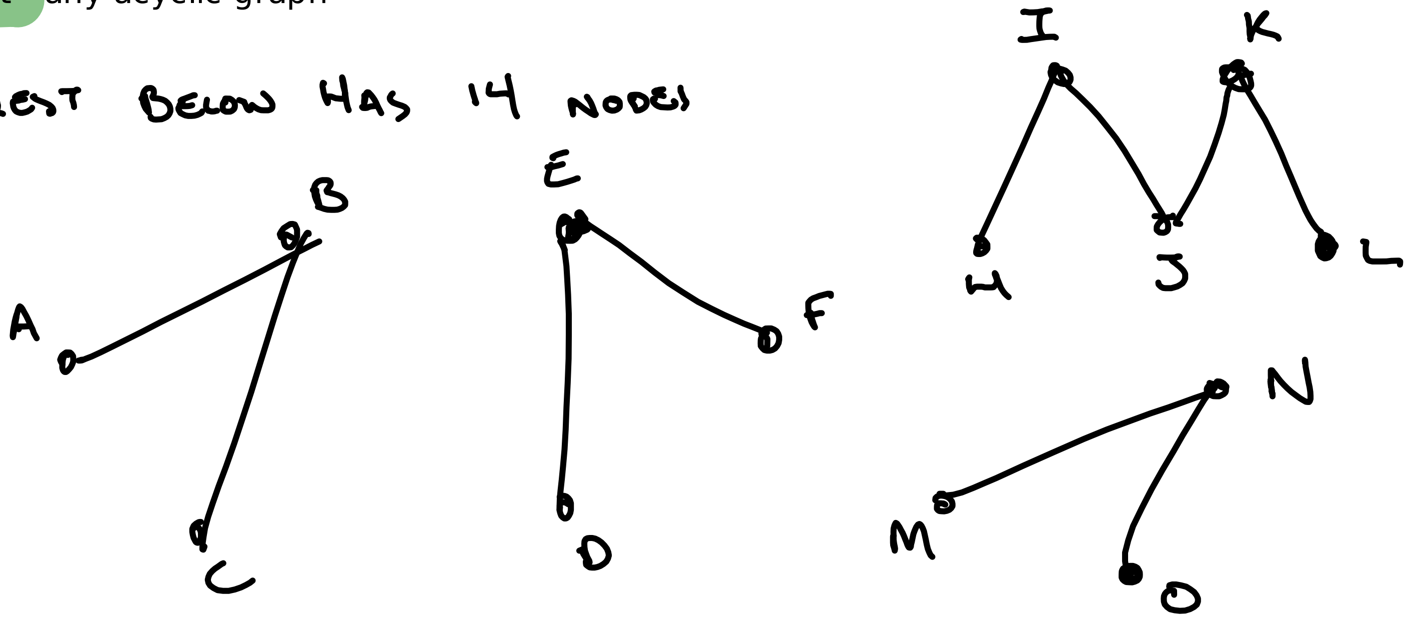


EACH CIRCLED SUBGRAPH IS A CONNECTED COMPONENT

tip: often thinking about a graph in terms of its connected components is fruitful for insights

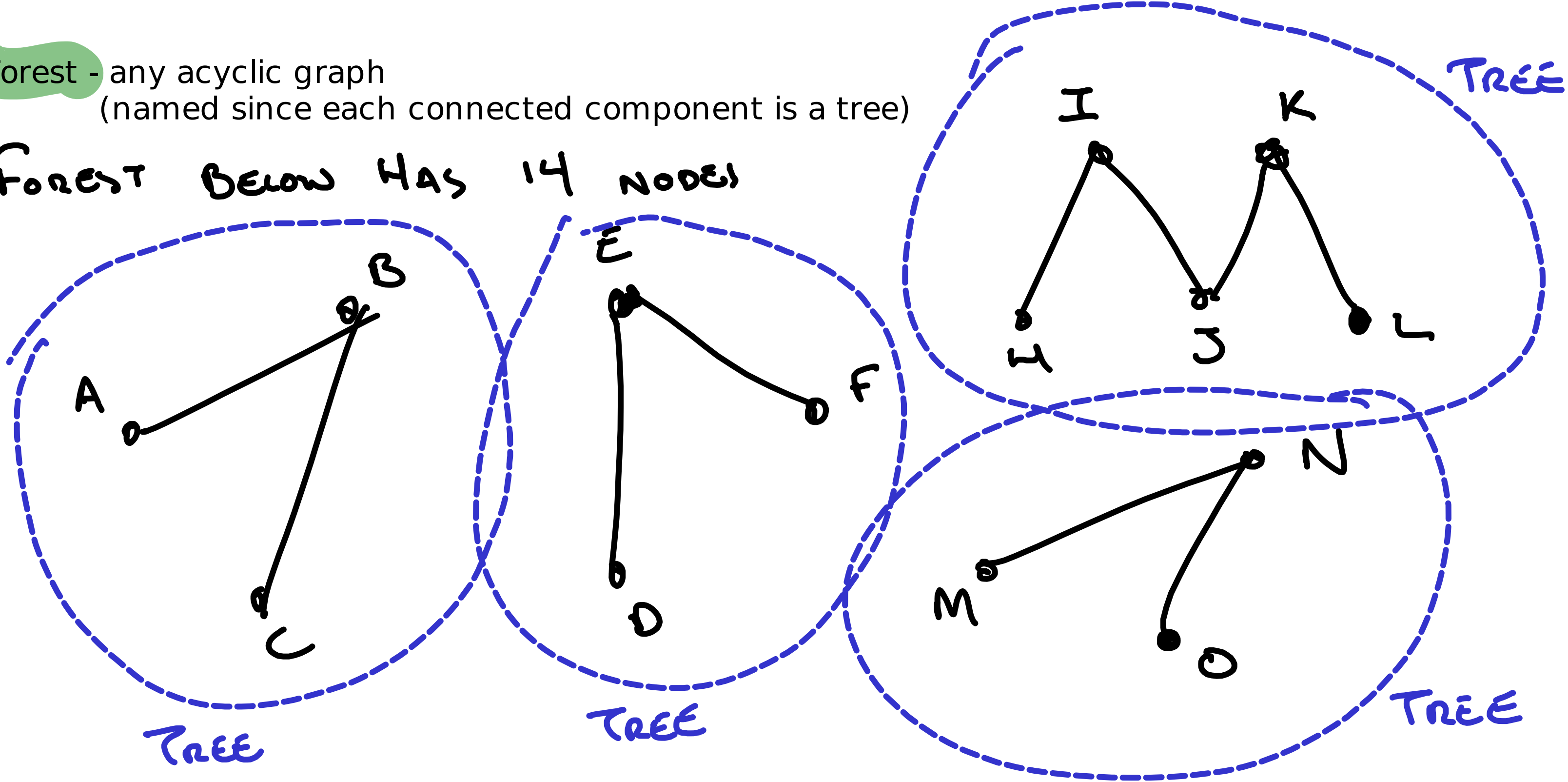
Forest - any acyclic graph

FOREST BELOW HAS 14 NODES



Forest - any acyclic graph
(named since each connected component is a tree)

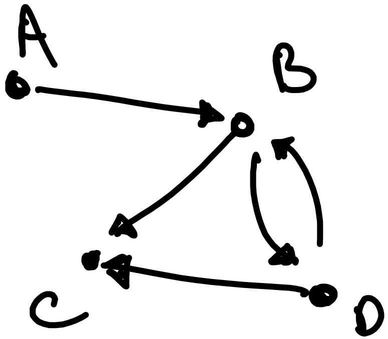
FOREST BELOW HAS 14 NODES



Special Graphs:

Directed

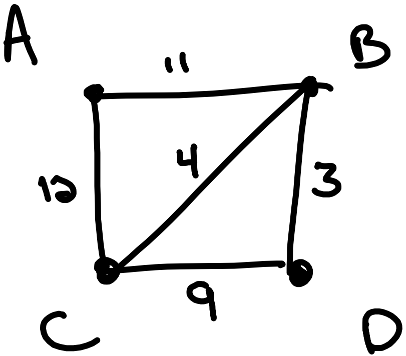
Each edge has a direction



"DAG": Directed Acyclic Graph

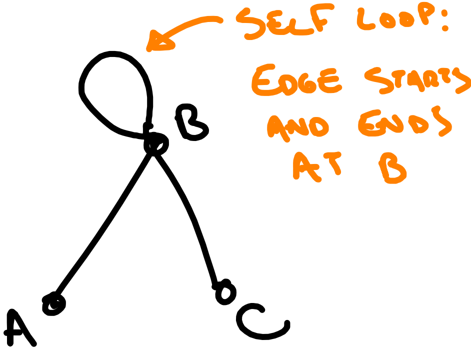
Weighted

Each edge has a weight



non-simple

Edge may start / end at same node



a simple graph has no such edge

ok, lets take a breather ... that was a lot of new language ...

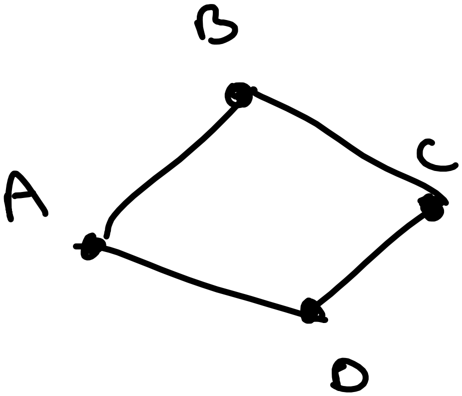
good news:

- only one more new graph vocab word today
- you needn't memorize anything, just take a peek back

not-so-good-news:

- graph language tends can have little inconsistencies per author
(e.g. is a node its own ancestor?)

Two nodes are neighbors if they are adjacent (there is an edge between them)
(note: definition assumes an undirected graph … edges have no direction)



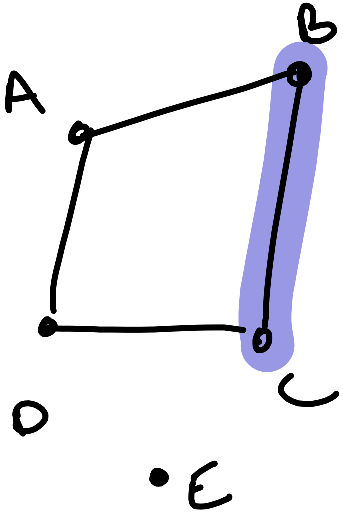
A, B ARE NEIGHBORS

A, C ARE NOT NEIGHBORS

Graph Representation (on a computer): List Representation

Goal: represent all nodes & edges of a graph

Approach: For each node, store a list of all neighbors (convention: alphabetize)



NEIGHBORS_A = [B, D]

NEIGHBORS_B = [A, C]

C IS NEIGHBOR OF B

NEIGHBORS_C = [B, D]

B IS NEIGHBOR OF C

NEIGHBORS_D = [A, C]

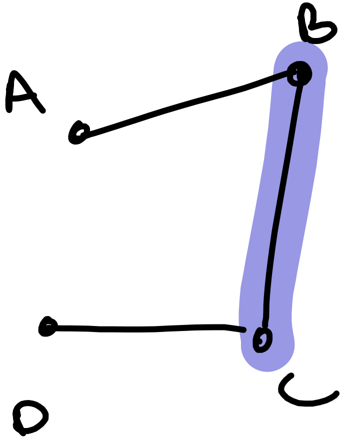
NEIGHBORS_E = []

Graph Representation (on a computer): Matrix Representation

Goal: represent all nodes & edges of a graph

Approach: Build |V| x |V| matrix (one row & col per node):

- 0 in row i and column j means node i and node j don't have edge between them
- 1 in row i and column j means node i and node j have edge between them



$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 0 & 0 \\ B & 1 & 0 & 1 & 0 \\ C & 0 & 1 & 0 & 1 \\ D & 0 & 0 & 1 & 0 \end{array}$$

THERE IS EDGE BETWEEN B,C

NO EDGE BETWEEN B,D

convention: alphabetize

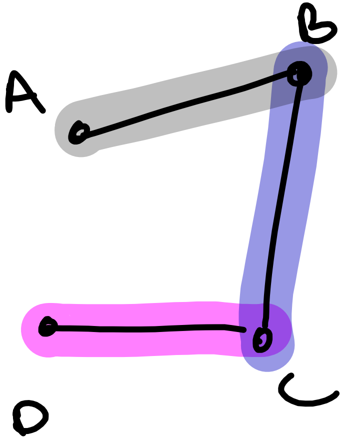
convention: a node is not its own neighbor

Graph Representation (on a computer): Matrix Representation

Goal: represent all nodes & edges of a graph

Approach: Build |V| x |V| matrix (one row & col per node):

- 0 in row i and column j means node i and node j don't have edge between them
- 1 in row i and column j means node i and node j have edge between them



	A	B	C	D
A	0	1	0	0
B	1	0	1	0
C	0	1	0	1
D	0	0	1	0

NOTICE: SYMMETRY

convention: alphabetize

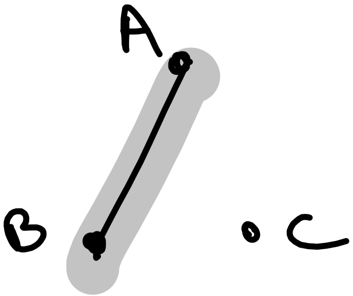
convention: a node is not its own neighbor

In Class Activity:

Given the one representation of the graph, give its representation as the other two forms.

Forms of representing a graph:
- picture (as is most common in the notes)
- list representation on computer
- matrix representation on computer

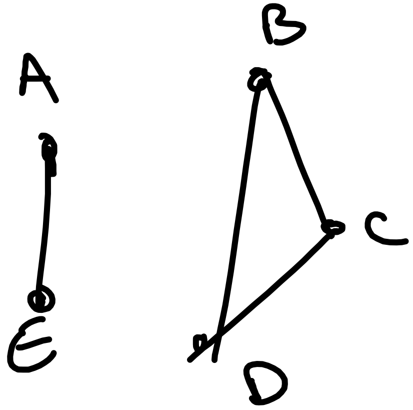
NEIGHBORS_A = [B]
NEIGHBORS_B = [A]
NEIGHBORS_C = [ ]



	A	B	C
A	0	1	0
B	1	0	0
C	0	0	0

	A	B	C	D	E
A	0	0	0	0	1
B	0	0	1	1	0
C	0	1	0	1	0
D	0	1	1	0	0
E	1	0	0	0	0

CONVENTION
NO NODE IS ITS OWN NEIGHBOR



A = [E]
B = [C D]
C = [B D]
D = [B C]
E = [A]

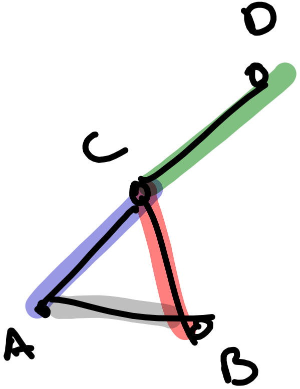
Graph isomorphism

high level: two graphs are isomorphic if they have same shape

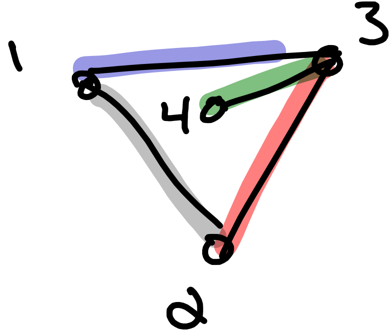
ISO - MORPHIC
↑ "SAME" ↑ "SHAPE"

intuition: two graphs are isomorphic when we can "rename" the nodes of one to get another

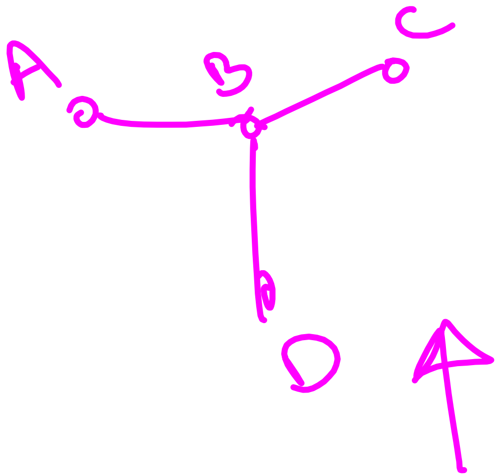
NODE RENAMING

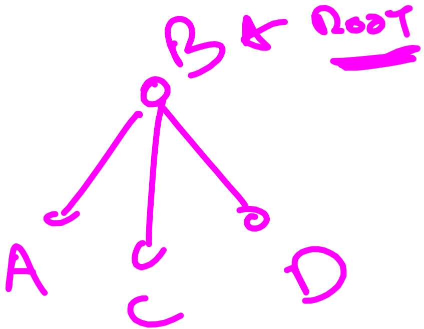


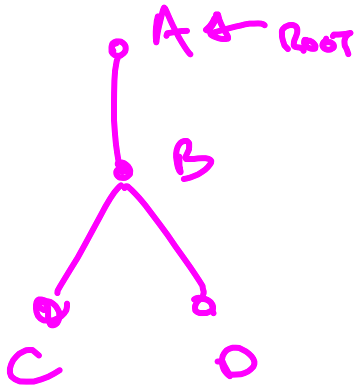
A = 1
B = 2
C = 3
D = 4



"rename": one-to-one correspondance (i.e. bijection)


A
B
C
D


B
ROOT
A
C
D


A
ROOT
B
C
D