CS 1800: day14

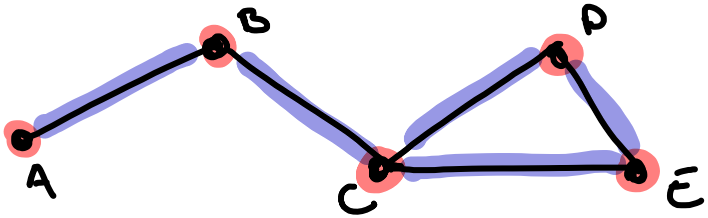
we'll start at 9:52 today :)

Admin:
- HW5 (probability) due Friday
- HW6 (graphs) released Friday
- Welcome to Raj (something going well / not so well in class? check in afterwards with him)

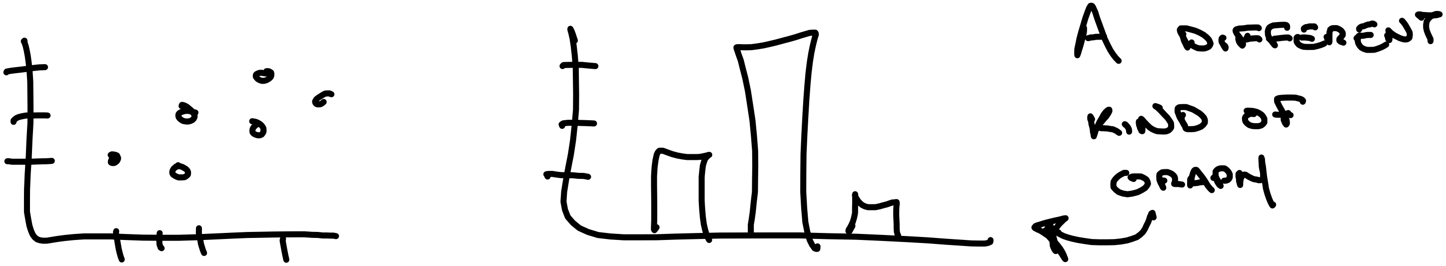
Content:
- graph definitions & anatomy
- graph representation
    - list of lists
    - adjacency matrix
- graph equivilence (isomorphism)

Whats a graph?

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



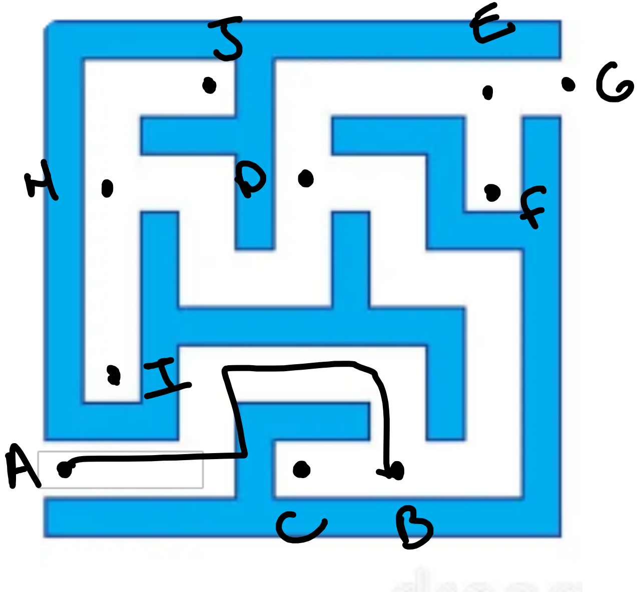
More commonly, folks use the word "graph" to mean figure (as below). This is a different kind of graph. Many tech types use the word "figure" to describe these, no universal convention



Graph: Whats it good for?

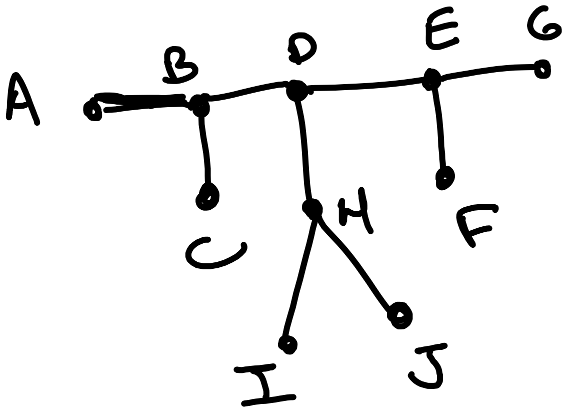
Graphs are wonderful for representing things. Often, representing clearly is a big help!

Example: represent a maze as a graph.



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

Edge = possible movement between
intersections

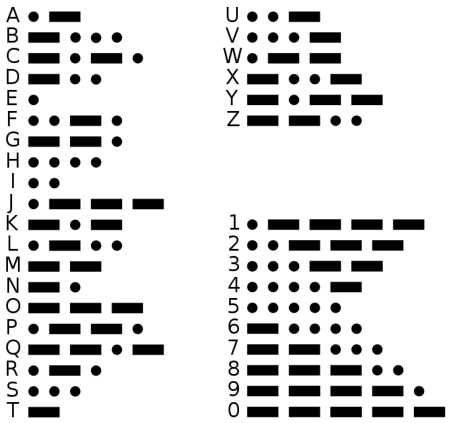


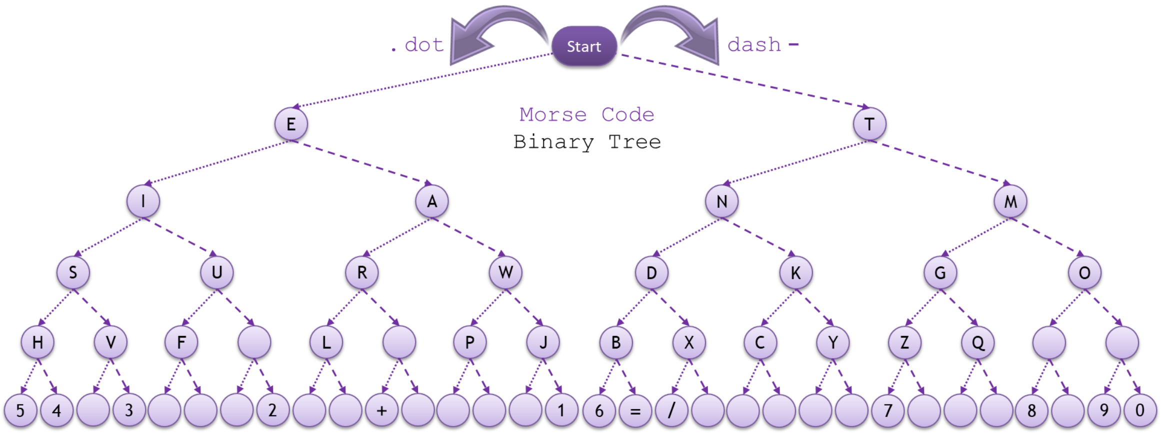
Graph: Whats it good for?

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

International Morse Code

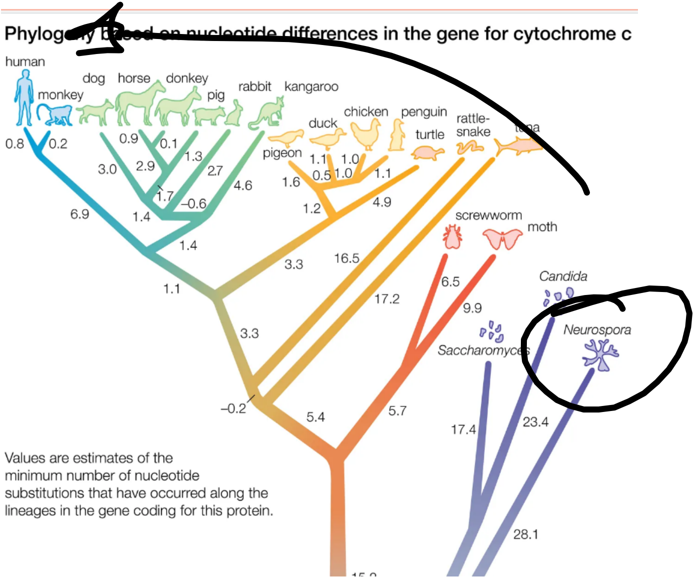
1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.





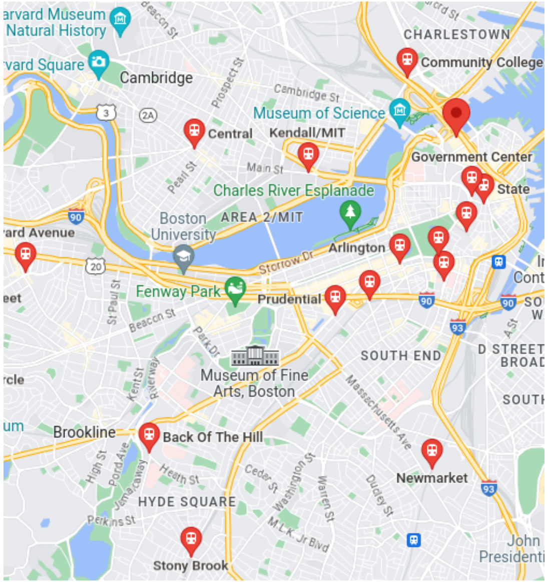
Graph: Whats it good for?

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!



Whats a graph?

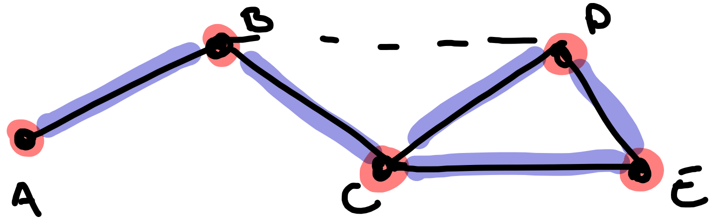
A set of nodes (also known as: vertex / vertices)

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

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

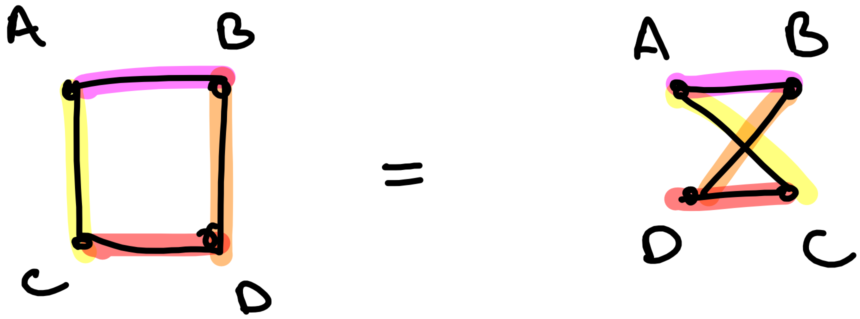
$$E = \{\{A,B\}, \{B,C\}, \{C,D\}, \{D,E\}, \{C,E\}\}$$

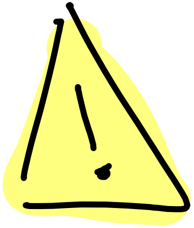
GRAPH REPRESENTED BY V, E TO LEFT:



$$\{B,D\} \notin E$$

GRAPHS CAN BE DRAWN DIFFENTLY
BUT ITS STILL SAME GRAPH





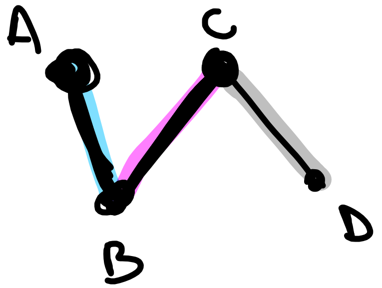
Warning:

The terms referring to graph features are often very intuitive.

(Be sure you don't rely exclusively on your intuition though, check definitions to ensure you're consistent ... your intuition can mislead you!)

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}

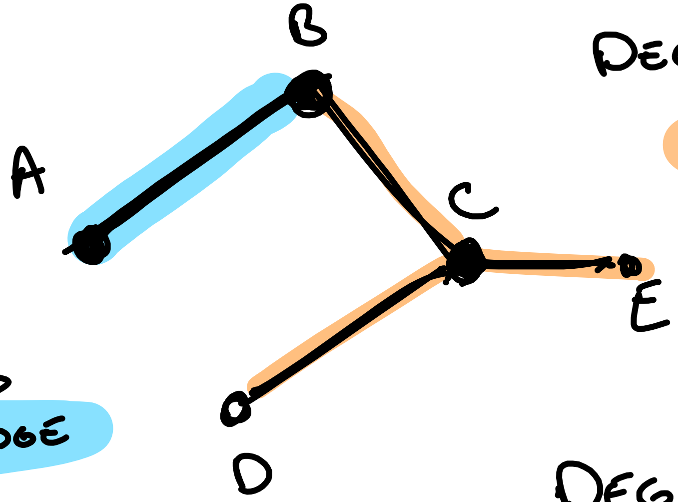
Not Adjacent: C, {A,B}

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

DEG(B)=2

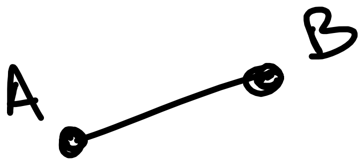
In Class Activity:

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

What is the relationship between the following values:

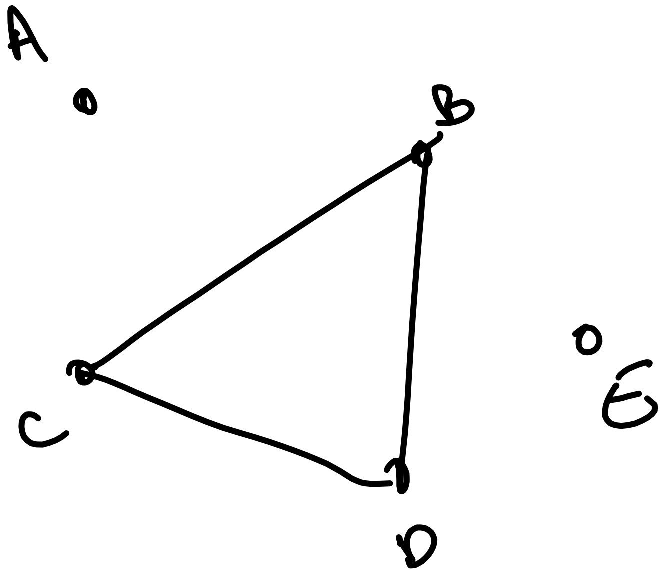
- the sum of degrees for all nodes
- the number of edges in the graph

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



$DEG(A) = 1$

$DEG(B) = 1$



$$\sum_{\text{node}} \text{DEG(NODE)} = 2 \cdot \#\ \text{EDGES}$$

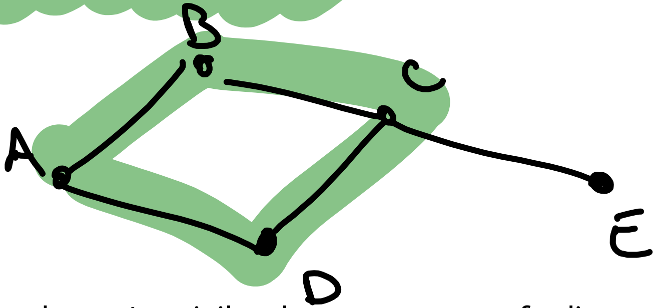
$0 = 0$

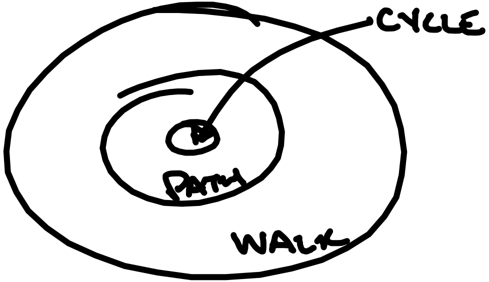
$2 = 1 \cdot 2$

$4 = 2 \cdot 2$

$6 = 3 \cdot 2$

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 SINCE B REPEATED

Path - a walk where each node is unique (except for possibly, the start / end node)

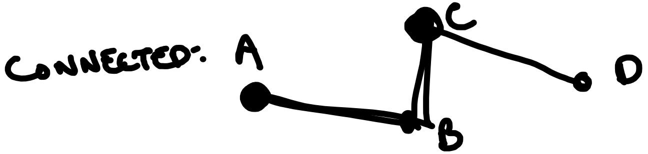
ABCE

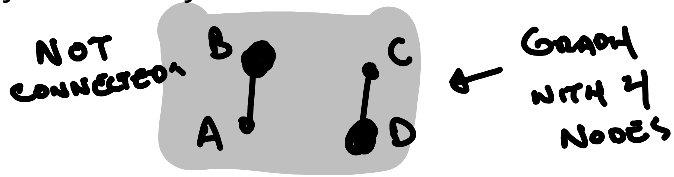
Cycle - a path which starts and ends at the same node

ABCDA

Trees! (a super useful construction)

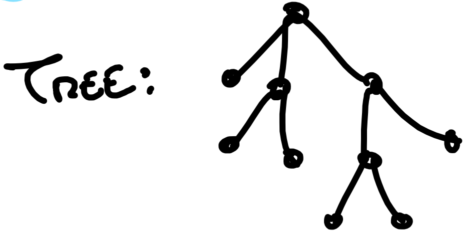
A graph is connected if there exists a path from every node to any other node





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

Tree - a connected graph without any cycles



In Class Activity

TREE
→ CONNECTED
→ ACYCLIC (NO CYCLES)

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

NODES = # EDGES + 1

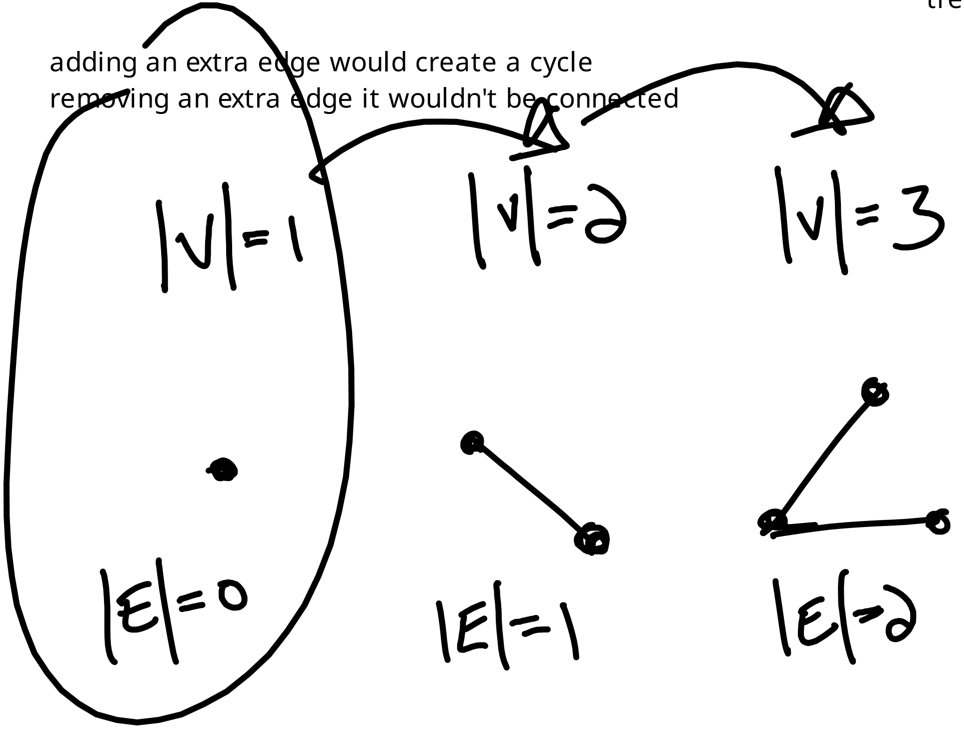
$|V| - 1 = |E|$

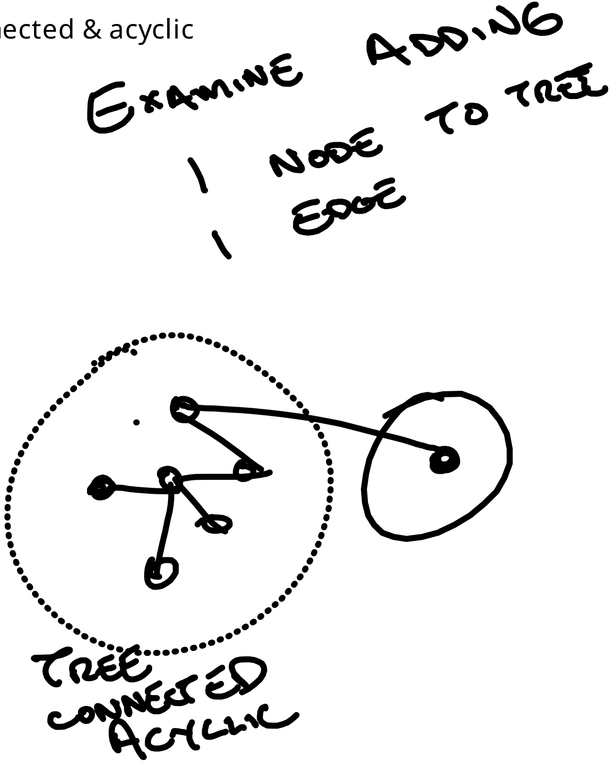
approach:
- draw some little examples
- make a conjecture (a guess as to the relationship)
- argue with your conjecture
- if you believe it, write out an explanation of why your conjecture is true

|V| - 1 = |E|

tree is connected & acyclic

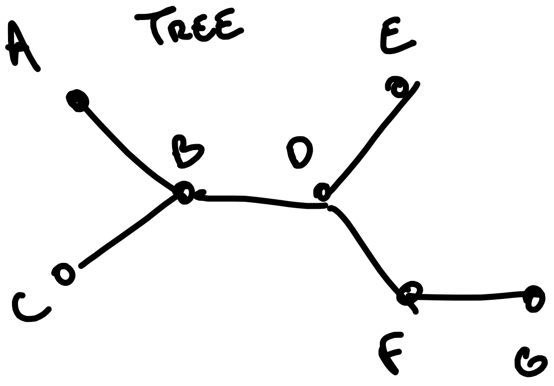
adding an extra edge would create a cycle
removing an extra edge it wouldn't be connected

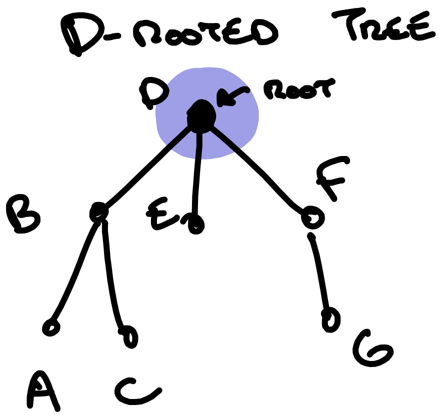


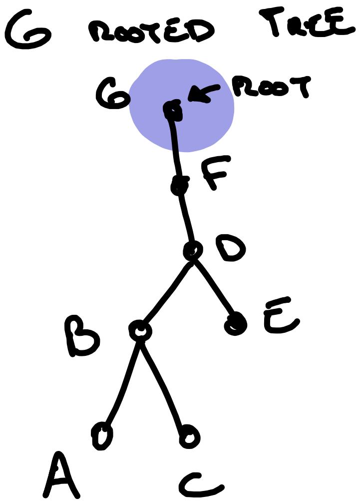


Rooted Trees

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







⚠ NOTICE: GRAPHS DIFFER ONLY BY ROOT

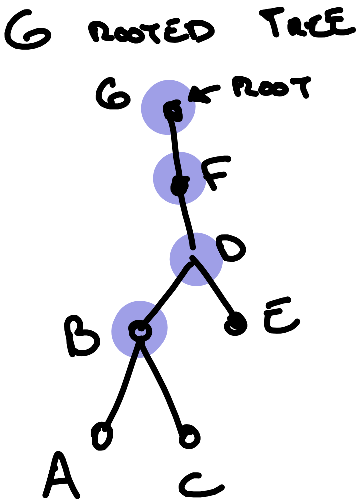
CONVENTION

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? ... it 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 whose parent is x

ex: {B, E} are children of D

a node is a leaf if it has no children:

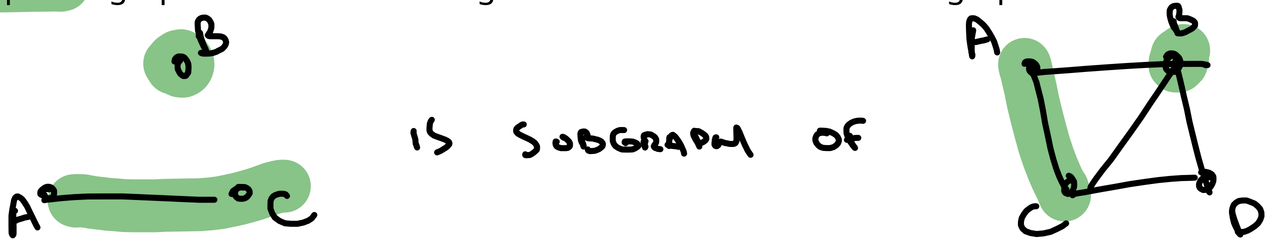
ex: A and C are leafs

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

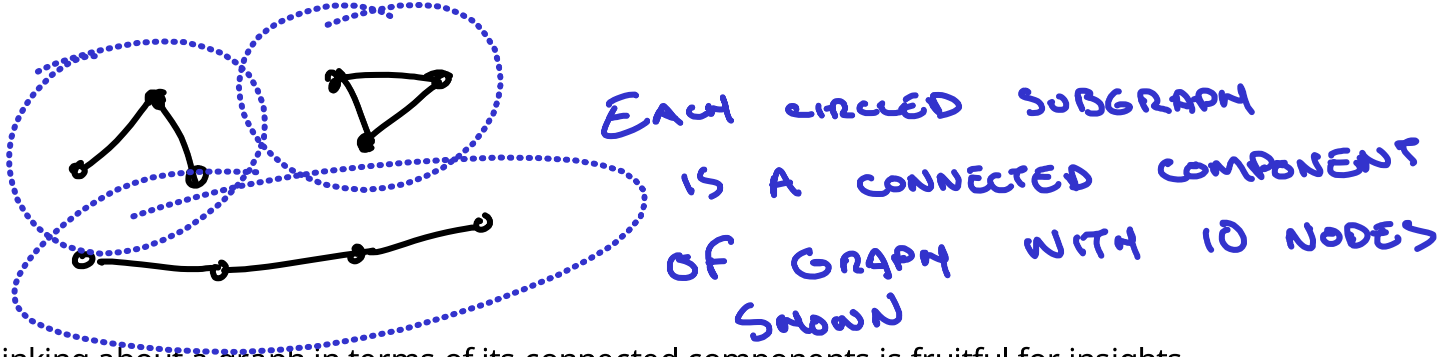
descendant of x: all nodes whose ancestor is x

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



Connected Component- a "largest" connected subgraph

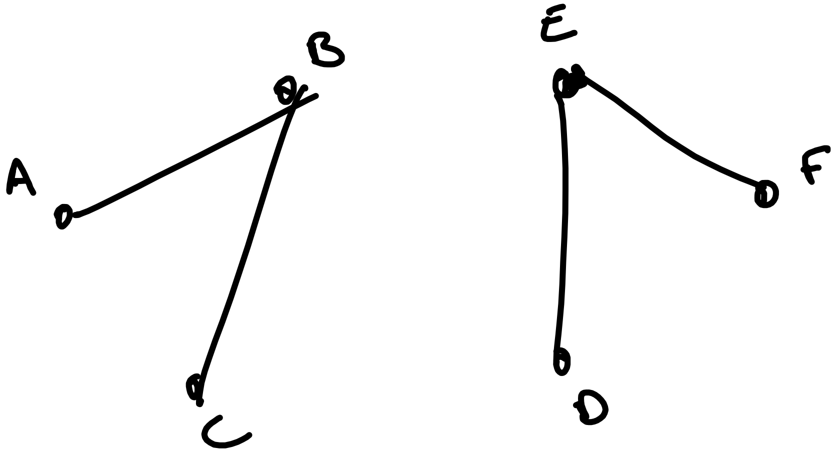
("largest" = not part of any larger connected subgraph)

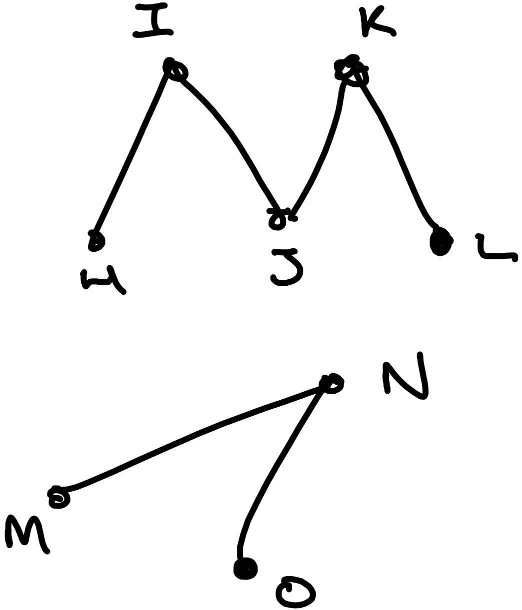


tip: thinking about a graph in terms of its connected components is fruitful for insights

Forest - any acyclic graph

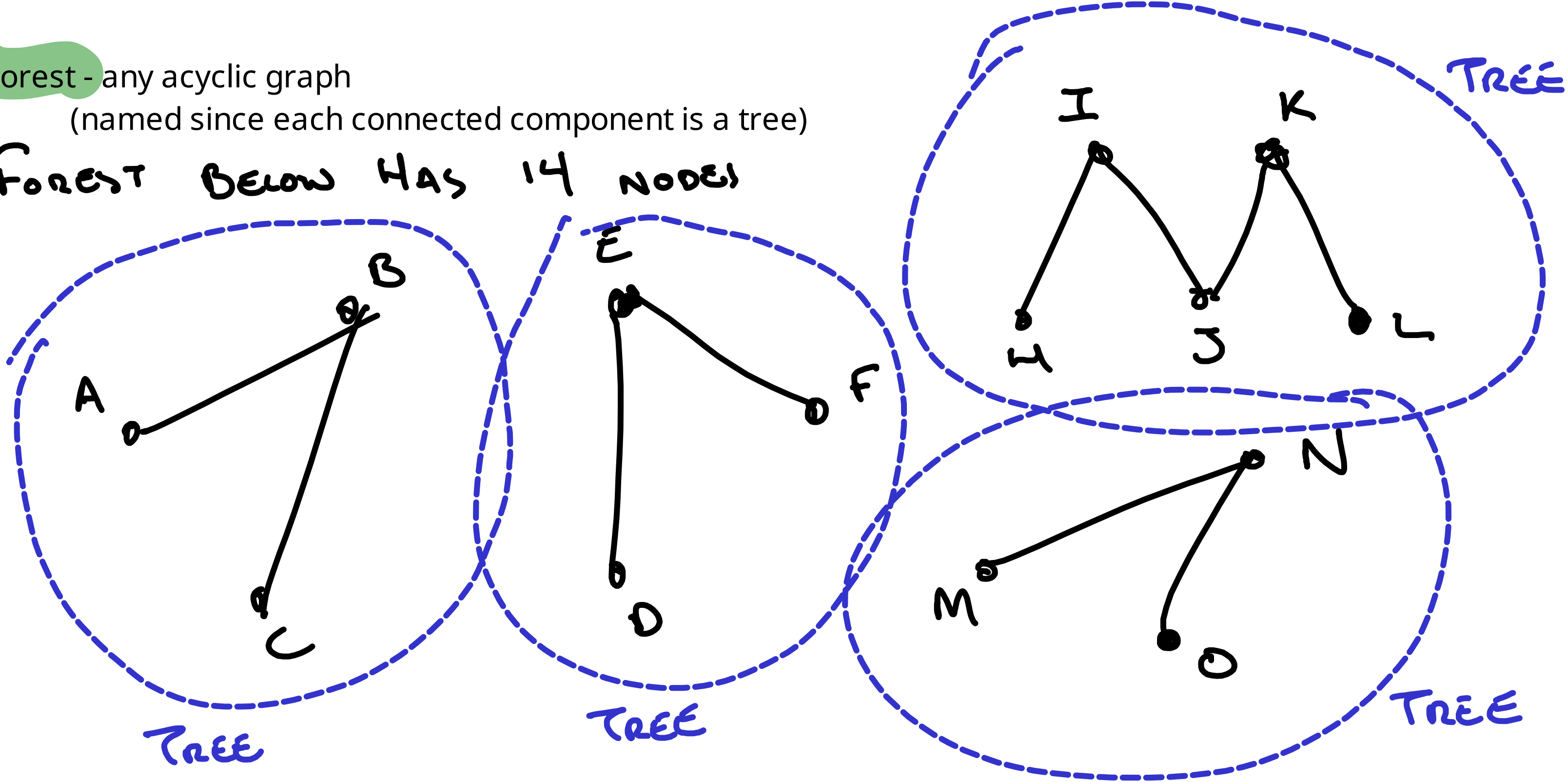
FOREST BELOW HAS 14 NODES





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

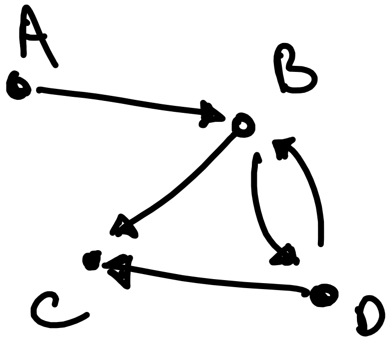
FOREST BELOW HAS 14 NODES



Special Graphs:

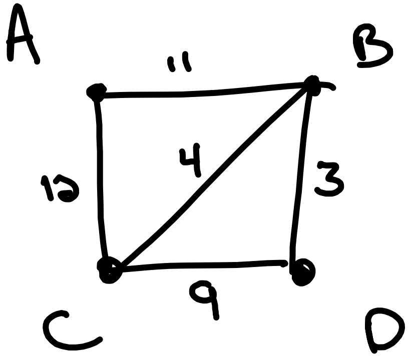
Directed

Each edge has a direction



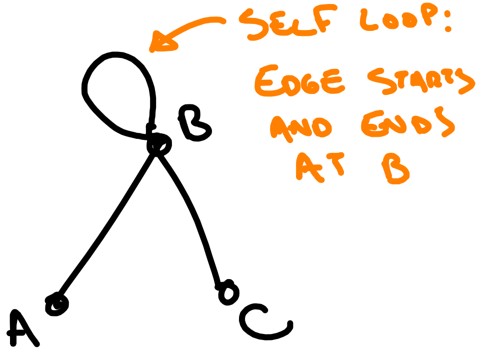
Weighted

Each edge has a weight



non-simple

Edge may start / end at same node



a simple graph has no such

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

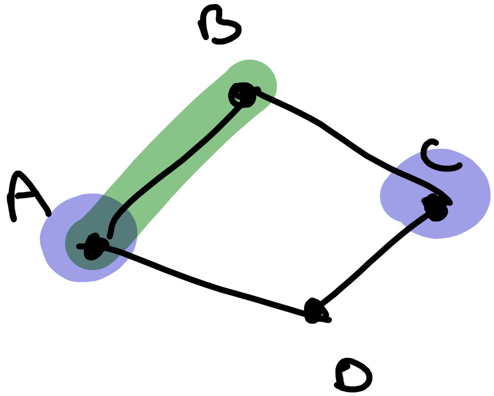
good news:

- 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)



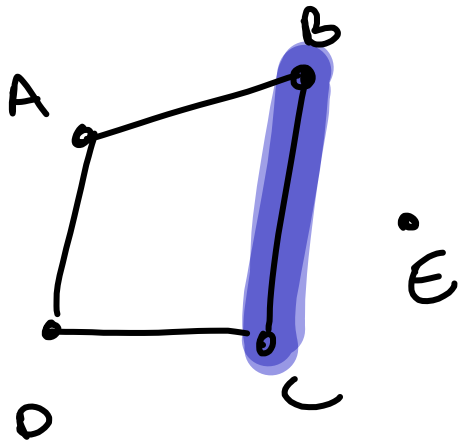
A, B ARE NEIGHBORS

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: order alphabetically)



NEIGHBORS_A = [B, D]

NEIGHBORS_B = [A, C] → C is NEIGHBOR OF B

NEIGHBORS_C = [B, D] → B is NEIGHBOR OF C

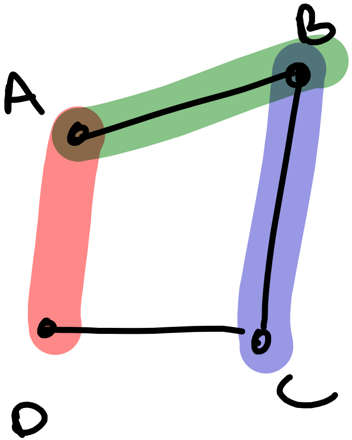
NEIGHBORS_D = [A, C]

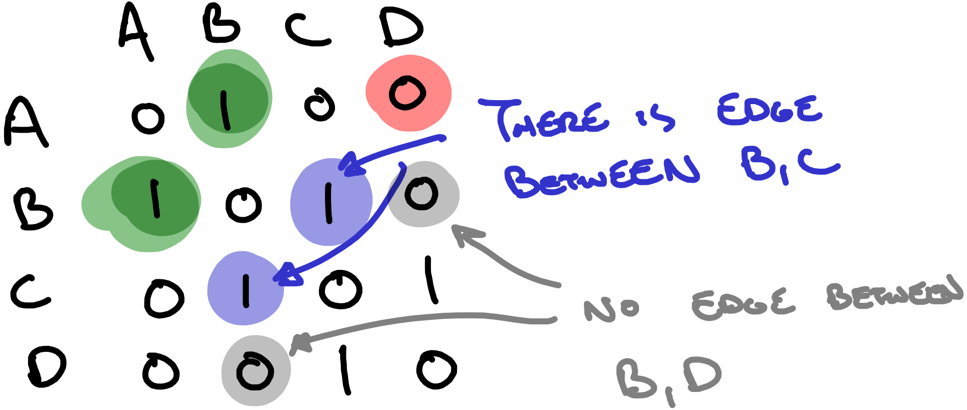
NEIGHBORS_E = [ ]

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



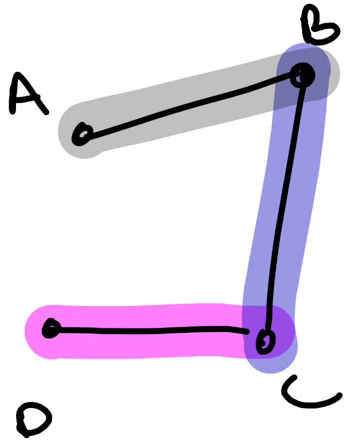


Graph Representation (on a computer): Matrix Representation

Goal: represent all nodes & edges of a graph

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

- 1 in row i and column j means node i and node j have edge between them (otherwise 0)



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

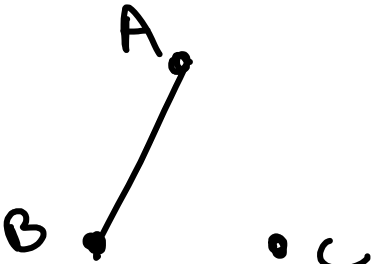
NOTICE: SYMMETRY

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

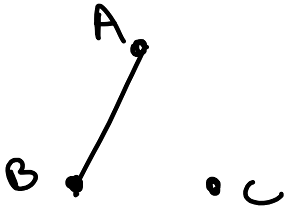


	A	B	C
A	0	1	0
B	1	0	0
C	0	0	0

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

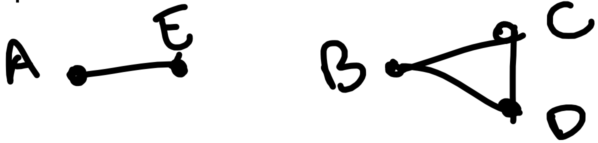
In Class Activity:

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



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

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

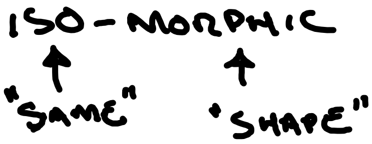


	A	B	C	D	E
A	0	0	0	0	1
B	0	0	1	1	0
C	0	1	0	1	0
D	0	1	1	0	0
E	1	0	0	0	0

neighbors_A = [E]
neighbors_B = [C, D]
neighbors_C = [B, D]
neighbors_D = [B, C]
neighbors_E = [A]

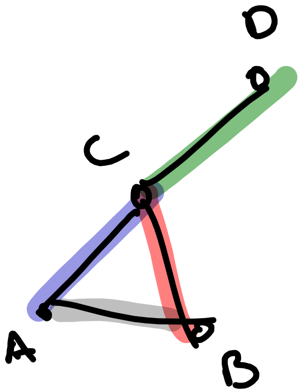
	A	B	C	D	E
A	0	0	0	0	1
B	0	0	1	1	0
C	0	1	0	1	0
D	0	1	1	0	0
E	1	0	0	0	0

Graph isomorphism



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

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



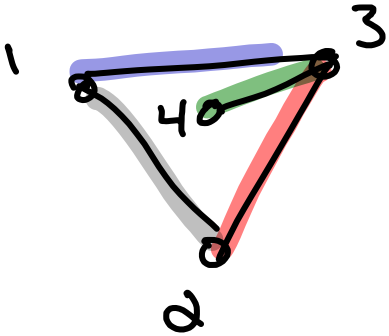
NODE RENAMING

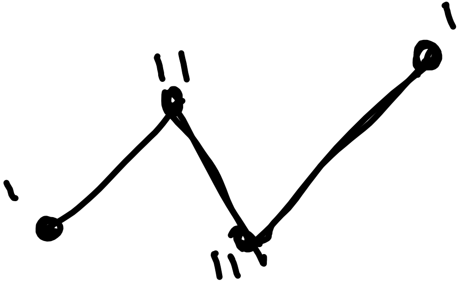
A = 1

B = 2

C = 3

D = 4





$$2 \cdot |E| = \sum_{n} DEG(n)$$