GRAPHS (REVIEW)
Thursday, October 18, 2018 5:12 PM

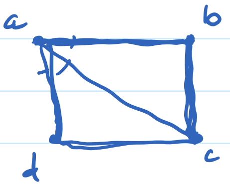
Def: A graph $G = (V, E)$ consists of a finite set of VERTICES and a set of EDGES which are <u>unordered</u> pairs of vertices

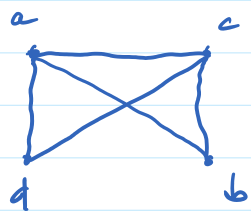
Ex: $G = (\underbrace{\{a, b, c, d\}}_{V}, \underbrace{\{\{a,b\}, \{b,c\}, \{c,d\}, \{d,a\}, \{a,c\}\}}_{E})$

Note: edge $\{a, b\}$ is the same as $\{b, a\}$ because pairs are UNORDERED)

Diagram representations

a, b, c
a, b, c, d, a
a, b, c, d, a, b, c





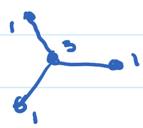
Definitions:
- Two vertices a and b are <u>adjacent</u> if $\{a, b\}$ is an edge
- A vertex a is <u>incident</u> to an edge e, if a is an endpoint of e
- The <u>degree</u> of a vertex v, $d(v)$, is the number of edges incident to v.

  Ex: $d(a) = 3$, $d(b) = 2$, $d(c) = 3$, $d(d) = 2$.

CLAIM (handshake): Sum of degrees in a graph $= 2 *$ number of edges

sum of degrees $= 3 + 2 + 3 + 2 = 10 = 2 * 5$

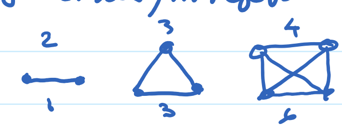
# of edges $= 5$

Ex: 

$3 + 1 + 1 + 1 = 2 * 3$

Proof: Each edge $\{a, b\}$ contributes 2 to the sum of degrees:
1 to $d(a)$ and 1 to $d(b)$
So total sum of degrees $= 2 *$ # of edges.

Def: In the context of graphs, n refers $|V|$, # of vertices, m refers to $|E|$, # of edges.



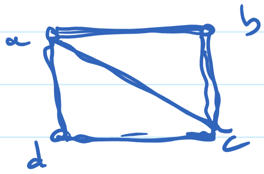
Relationships between n, m: a n-vertex graph has between 0 and $\frac{n(n-1)}{2}$

$\binom{n}{2}$ n choose 2 $\frac{n(n-1)}{2!} = \frac{n!}{2!(n-2)!}$

Def: A **path** in a graph is a sequence of vertices $v_1, v_2, \ldots, v_k$ such that $\{v_i, v_{i+1}\} \in E$ $1 \le i < k$.

A **simple path** is a path whose vertices are unique

A **cycle** is path whose first and last vertices are the same



a–b–c–a is a cycle

a–b–c–d–a is a cycle

a–b–c–d–a–b–c–a

A **simple cycle** is a cycle with no repeated vertices except for the first and last.

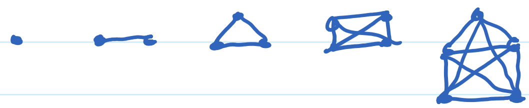
NOTE: by definition, a cycle must have at least 3 edges.

SPECIAL CLASSES

1) complete graphs: all possible edges are present.

n = 1 2 3 4 5

C_n



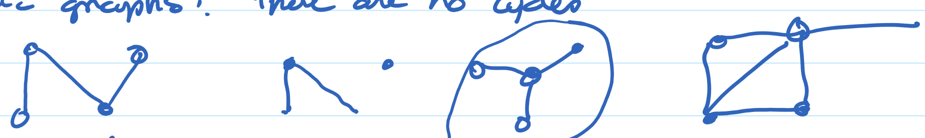
. . . . .

2) connected graphs: there is a path between any pair of vertices

Ex.:

connected / 1 connected — disconnected / has 2 connected components — 4 connected components

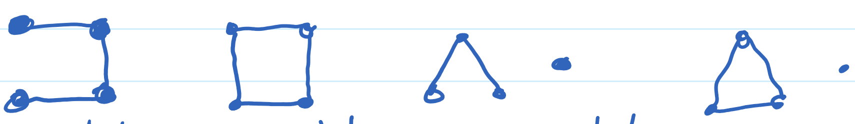
3) acyclic graphs: there are no cycles



acyclic — acyclic — acyclic — cyclic

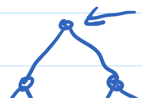
Connectedness and acyclicity are independent notions

tree →



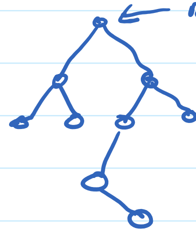
connected acyclic (tree) — connected cyclic — disconnected acyclic — disconnected cyclic

4) Trees: acyclic & connected graph



← rooted

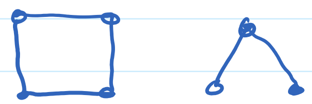
4) Trees : acyclic & connected graph
(unrooted)



<u>THEOREM</u> : the following are equivalent : Let T be a graph with n vertices and m edges.

1) T is acyclic and connected
2) T is acyclic and $n = m+1$
3) T is connected and $n = m+1$
4) There is a UNIQUE path between any pair of vertices in T.

<u>Ex</u> :

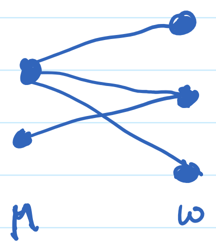


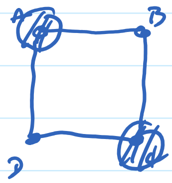
$n = 7$
$m = 6$

$n = m+1$ but this graph is not a tree

5) BIPARTITE : if $V = M \cup W$, $M \cap W = \emptyset$
and $E \subseteq \{\{m, w\} : m \in M, w \in W\}$

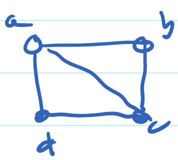
<u>Ex</u> :





GRAPH REPRESENTATIONS

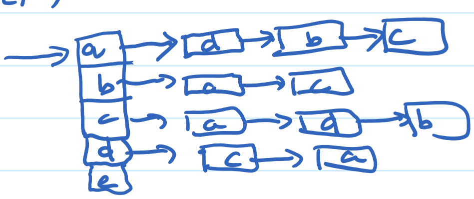
1) Adjacency Matrix : matrix $M[i][j] = \begin{cases} \text{true} & \text{if } \{i,j\} \in E \\ \text{false} & \text{o.w} \end{cases}$



	a	b	c	d
a	F	T	T	T
b	T	F	T	F
c	T	T	F	T
d	T	F	T	F

<u>Advantage</u> : is_edge(i, j) in constant time
<u>Disadvantage</u> : wasted space
$\Theta(n^2)$

2) Adjacency lists : a linked list of adjacency vertices for each vertex
(DEFAULT)

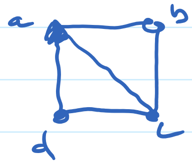


<u>Advantage</u> : space - efficient
# of nodes : $n + 2m$ $\Theta(n+m)$
<u>Disadvantage</u> : is-edge(i, j) may take $d(i)$ steps

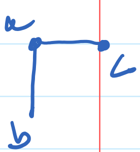
<u>RECOGNIZING SPECIAL GRAPHS</u>

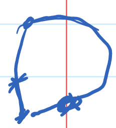
1) complete: check that there are n-1 neighbors in each list
Run time: $\Theta(n + 2m)$

2) <u>Connected</u> : use Breadth-First Search (BFS)



s = start vertex
identify vertices ONE edge away from s
TWO edges ..
THREE ..





Ex: start vertex = a

~~a d b c e~~

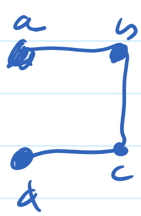
visited

a	b	c	d	e
T	T	T	T	T

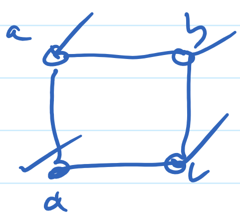
```
Q.push(s);
visited[s] = true;
while (!Q.empty)
{   f = Q.front();
    Q.pop();
    for each vertex w in Adj[f]
        if !visited[w]
        {   visited[w] = true; Q.push(w);
        }
}
```

b - d

3) Acyclic : BFS ; there is a cycle if a vertex has 2 visited neighbors



~~a b c d~~

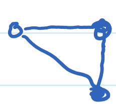


~~a b d~~ c

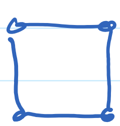
cycle

4) tree : acyclic & connected

5) bipartite :

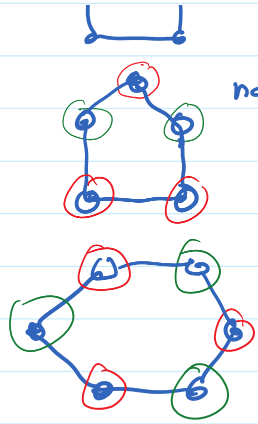


not bipartite

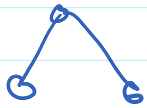


bipartite

not bipartite

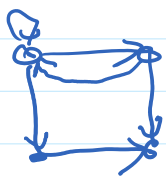


not bipartite



Theorem: G is bipartite if and only if G has no cycles of odd lengths.

DIRECTED GRAPHS: edges are ORDERED PAIRS



WEIGHTED GRAPHS / DIGRAPHS