L17 – Graph Concepts

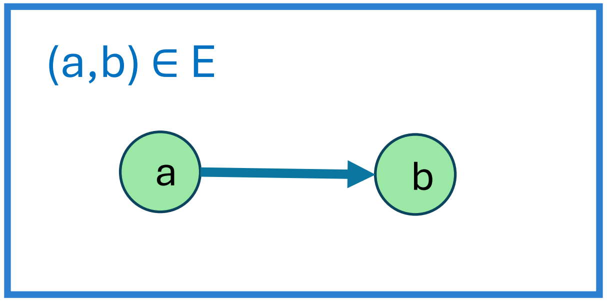
7/19/2024

Definitions

- Graph is not a picture or a chart
- Mathematical structure based in sets and relations/functions
- $G = (V, E)$ where
  - V is a set of vertices (nodes)
  - E is a set of edges (arcs)
- E is a set of ordered pairs $E = \{ (a,b) \mid a \in V \land b \in V \}$
  - directed graph
- E is a set of sets, $E = \{ \{a,b\} \mid a \in V \land b \in V \}$
  - undirected graph
- We let edges represent (model) different things
  - Road from a to b, a is parent of b, a employs b, a must be taken before b

Directed Graph

- Adjacent
- We say b is adjacent to a iff $(a,b) \in E$
- Directed graph ("digraph")
- Edges in E are directional
- Each pair in E is ordered
- Draw with arrows

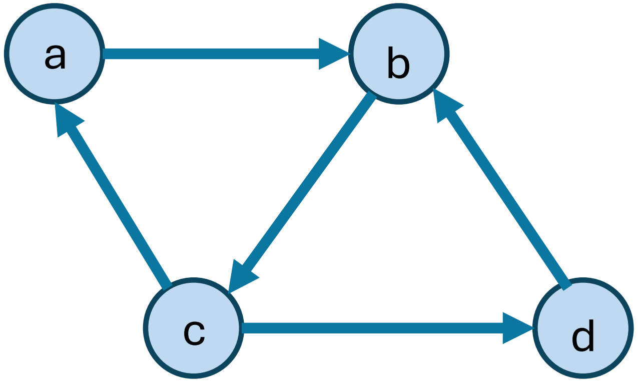


V = { a, b, c, d }

E = { (a,b), (b,c), (c,a), (c,d), (d,b) }

so

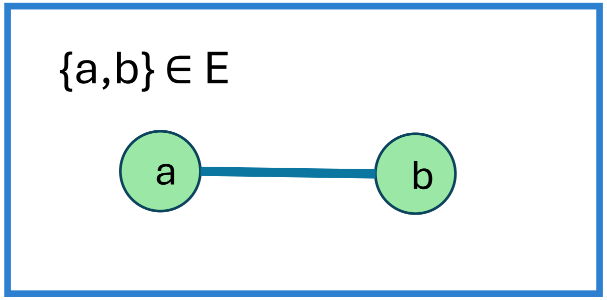
G = ({ a, b, c, d },
{ (a,b), (b,c), (c,a), (c,d), (d,b) })



Undirected Graph

- Edges have no direction
- If b is adjacent to a, then a is also adjacent to b
- Elements in E are sets, not ordered pairs

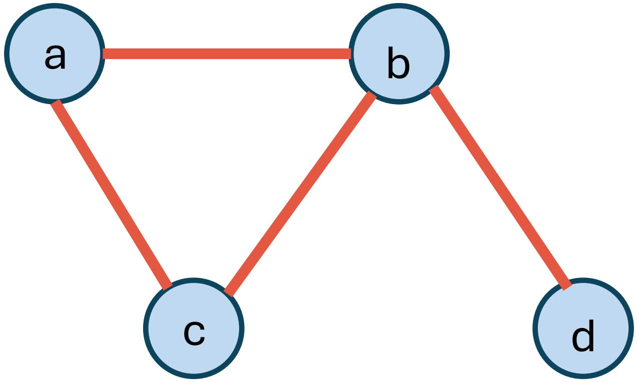
We say
a is adjacent to b, AND
b is adjacent to a



$V = \{ a, b, c, d \}$

$E = \{ \{a,b\}, \{a,c\}, \{b,c\}, \{b,d\} \}$

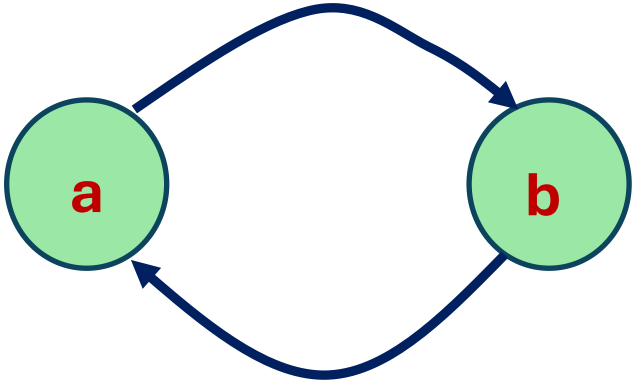
$G = (V, E)$



Symmetry in Digraph

Symmetry: $(a,b) \in E \wedge (b,a) \in E$

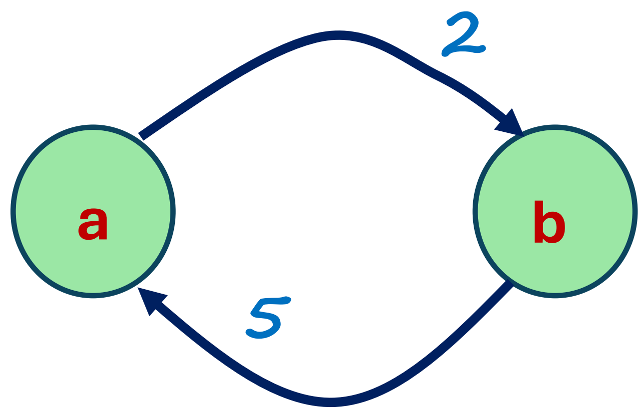
Not so here - only one edge



Not the same as this

Weighted Edges

Some problems may need a "weight" associated with each edge



E = { (s,d,w) |
s∈V ∧ d∈V ∧ w∈Int }

- Or weights might be real numbers
- Unweighted graphs can be thought of as having weight of 1 on each edge

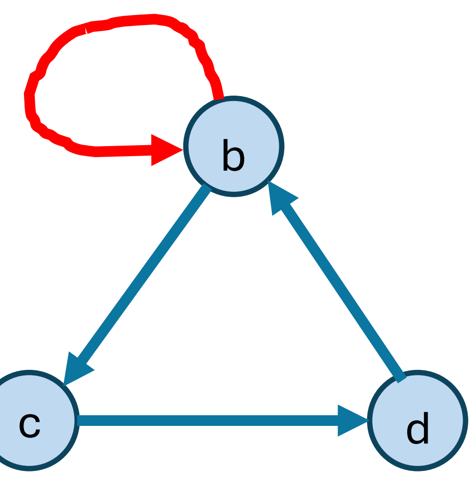
Path

- Path
  - Sequence of vertices $n_1\ n_2\ n_3\ ...n_k$ where $(n_i, n_{i+1}) \in$ E for $1 \leq i < k$
- Path Length
  - $k - 1$, the number of edges in the path
  - Every vertex has a path of length 0 to itself
  - This is not the same as *(b,b) ∈ E*

Here (b, b) ∈ E

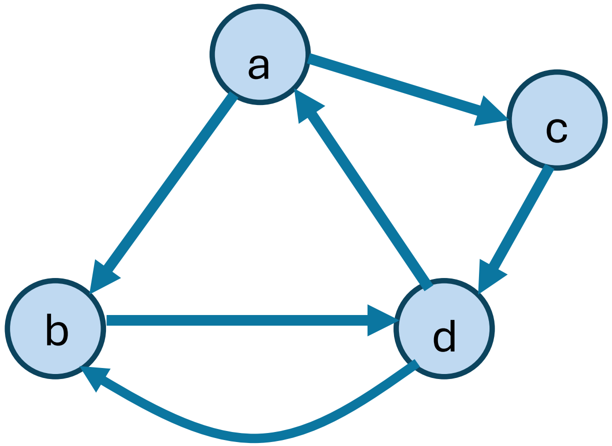
So there is a path b to b

that has length 1



Simple Cycle

- Simple Path
  - Sequence of vertices $n_1\ n_2\ n_3\ ...n_k$ where $n_j \neq n_i$ for distinct *i, j* (except $n_1 = n_k$ is ok)
  - a, b, d   is simple path
  - a, b, d, a   is simple path
  - a, b, d, a, c is *not* simple
  - a, c   is simple
- Cycle (in digraph)
  - Path of length ≥ 1 where $n_1 = n_k$
  - Starts and ends on same node
  - a, b, d, a   is a cycle (simple cycle, length 3)
  - b, d, a   is *not* a cycle
  - b, d, a, c, d, b is a cycle (length 5, but not simple)

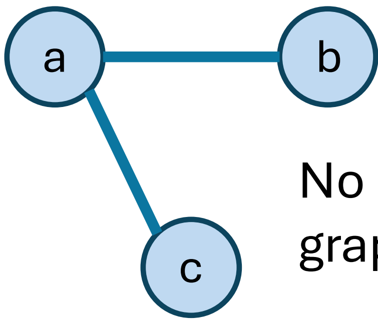


Cycle in Undirected Graph

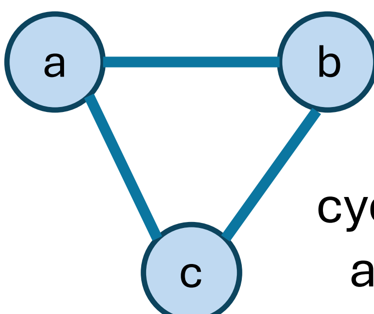
- We require the **edges** to be distinct
- If {a,b} ∈ E, there is not necessarily a cycle between a and b
  - If a, b, a is a cycle, this would imply an edge (a,b) and another edge (b,a) (path length 2)
  - But {a, b} is one edge, the same edge

Another reason these are technically not the same

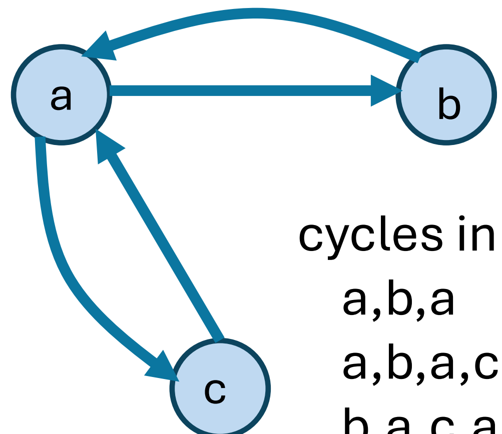
Cycle in Undirected Graph



No cycles in this graph



cycles now:
- a,b,c,a
- b,c,a,b
- etc.



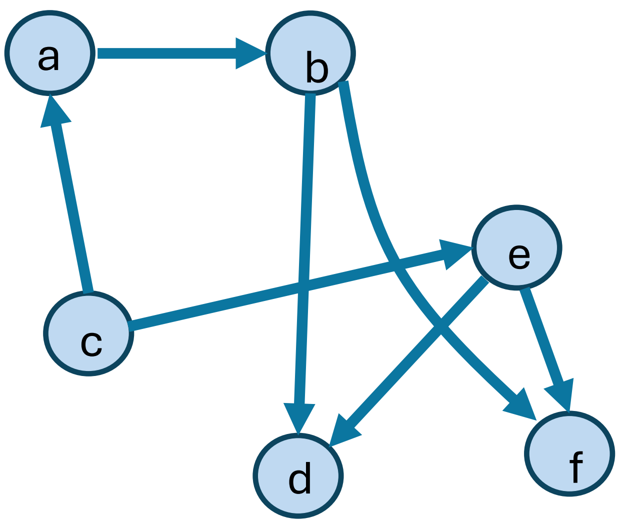
cycles in this one:
- a,b,a
- a,b,a,c,a
- b,a,c,a,b
- etc.

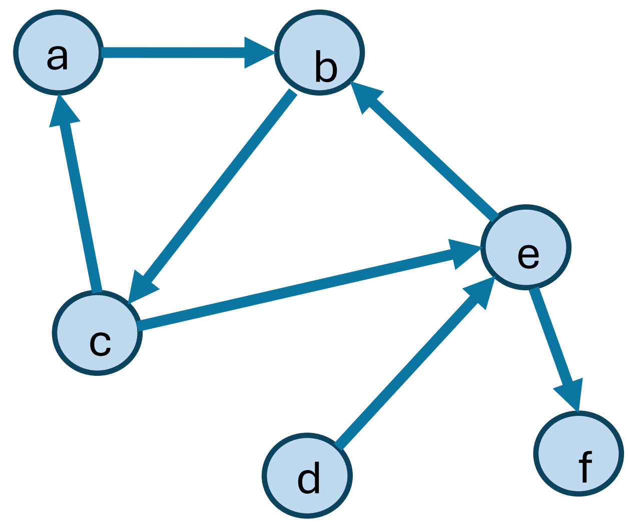
DAG example

- DAG
    - Directed Acyclic Graph
    - Special form used in many problems

Directed edges
No cycles

DAG



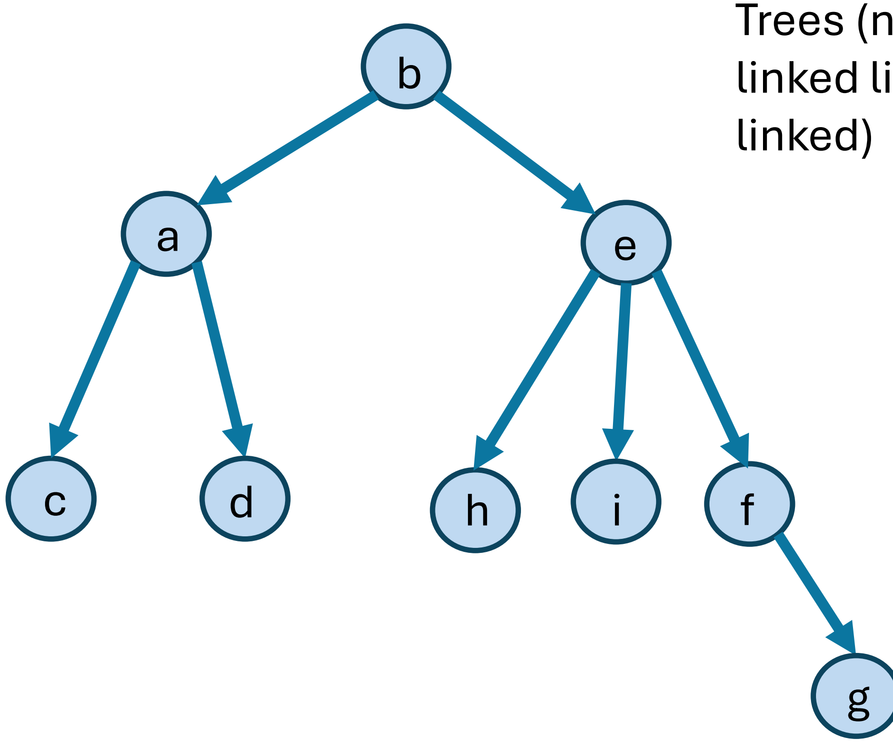


Directed edges
But cycles

Not a DAG

DAG

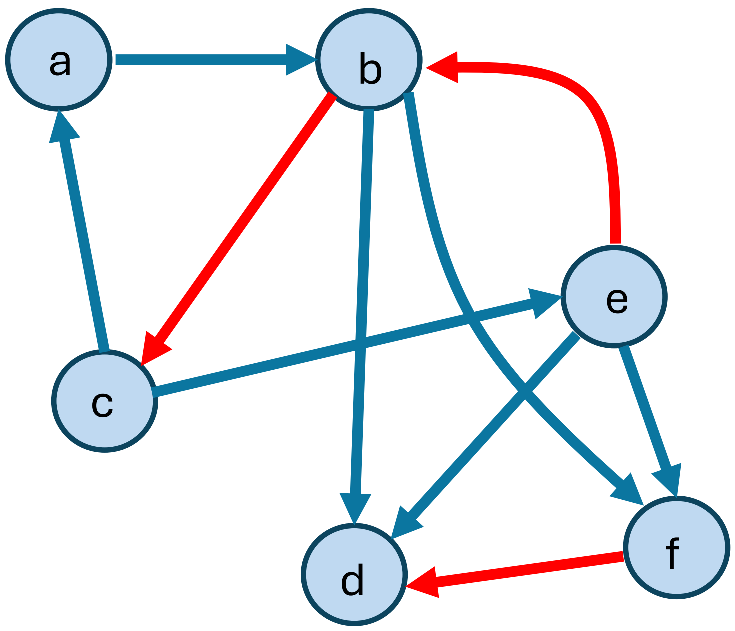
- We have already been using DAG's
- Where?



Trees (no parent pointer), linked lists (not doubly linked)

Graph algorithm – cycle detection

- How can we detect whether there are cycles?

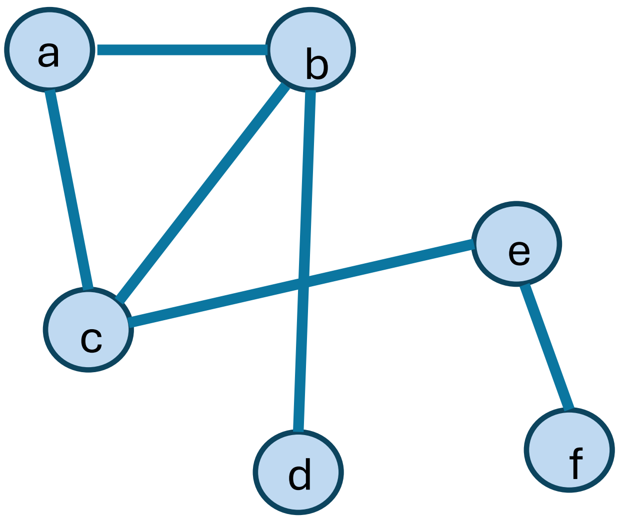


Now?	No, DAG
Now?	No, DAG
Now?	No, DAG
Now?	Yes

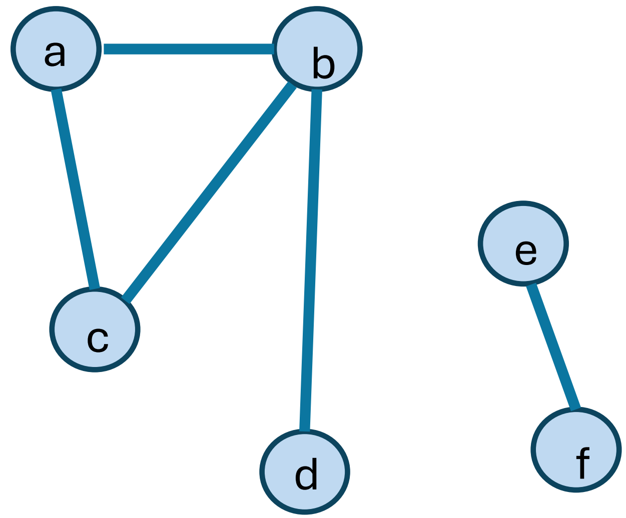
Graph Algorithm:
For each vertex v {
trace paths from v
see if you revisit a node on the path
each path must end or revisit (why?)
}

Connected - Undirected

- Connected
  - Has a path from every vertex to every other vertex



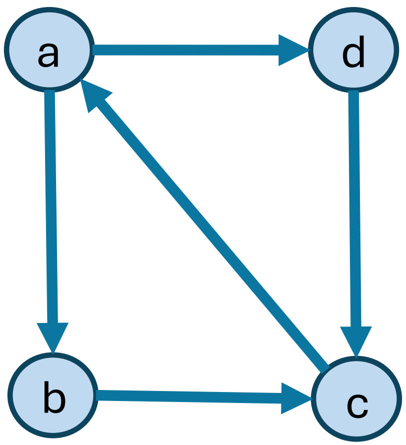
connected



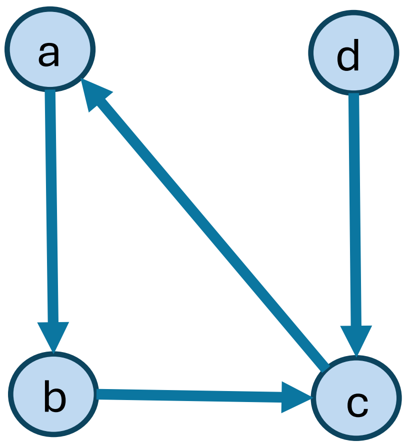
Not connected

Connected - Directed

- Strongly Connected
  - Has a path from every vertex to every other vertex



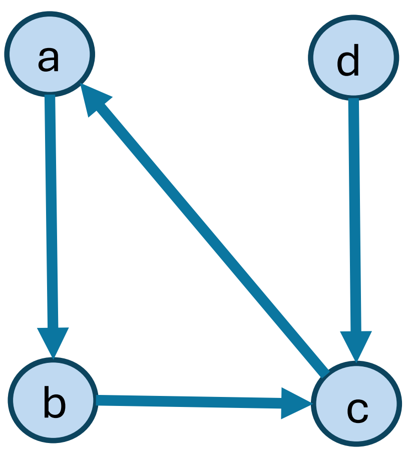
Strongly connected



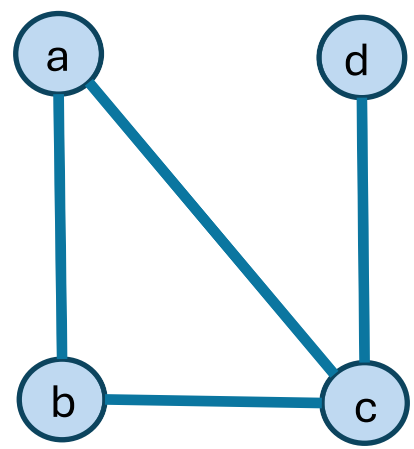
Not strongly connected

Connected - Directed

- Weakly Connected
    - Underlying undirected graph is connected



Underlying undirected graph

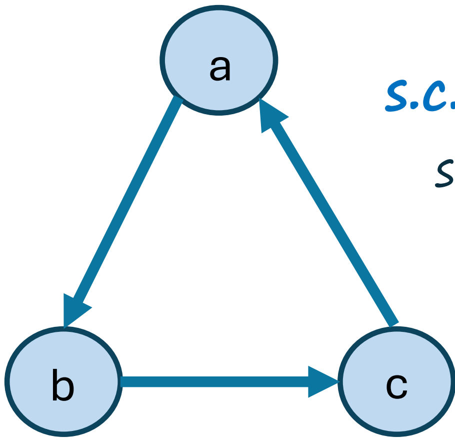


Not strongly connected

This is weakly connected

this is connected, so

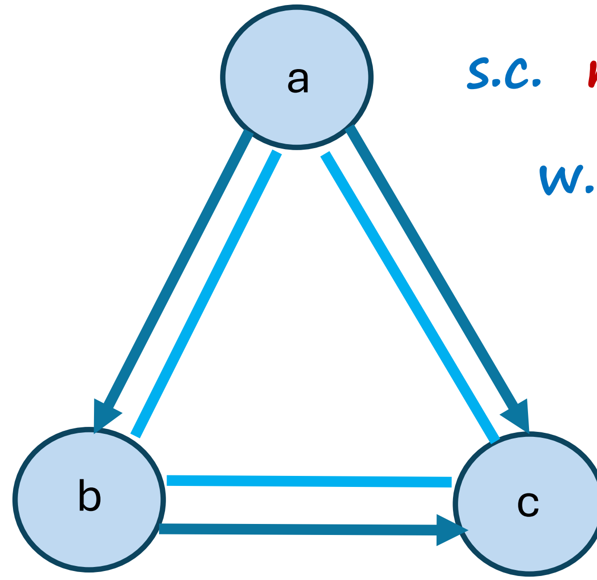
More Examples



s.c. yes

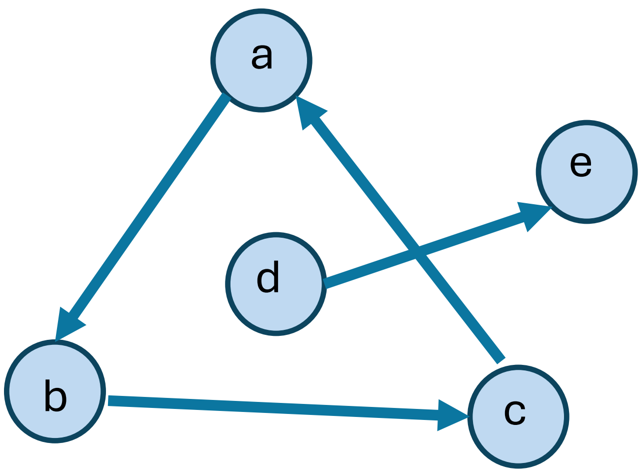
so also

w.c. yes



s.c. no

w.c. yes



w.c. no

so also

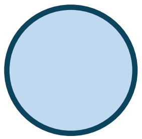
s.c. no

we just say

not connected

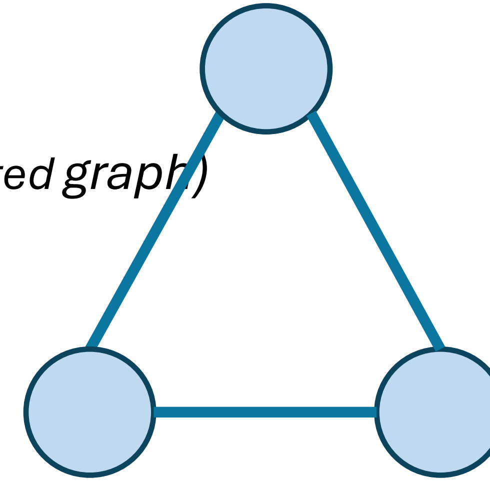
Complete Graph

- An edge between any two distinct vertices
- $\{n_i, n_j\} \in E$ for every $n_i, n_j \in V$, and $i \neq j$ *(undirected graph)*

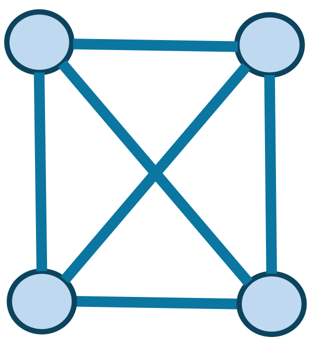


K1

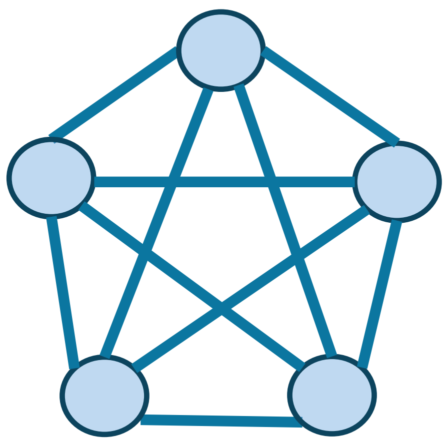
K2



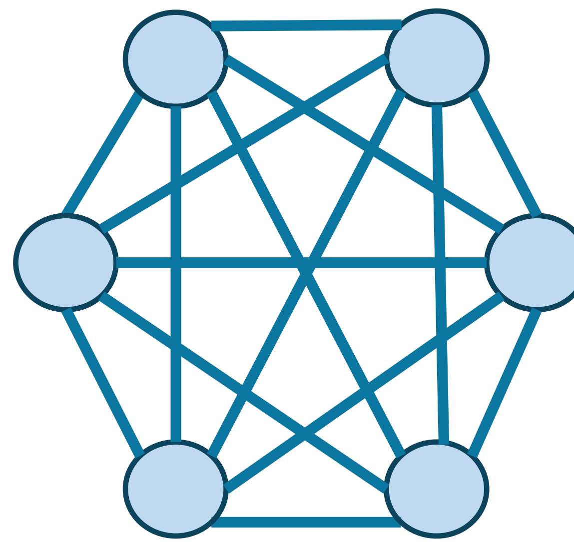
K3



K4



K5



K6

How many edges in an undirected complete graph?

- How many edges are in an undirected complete graph with v vertices?

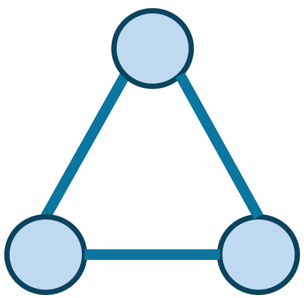
v(v-1)/2

Sum of natural numbers

What about digraph?

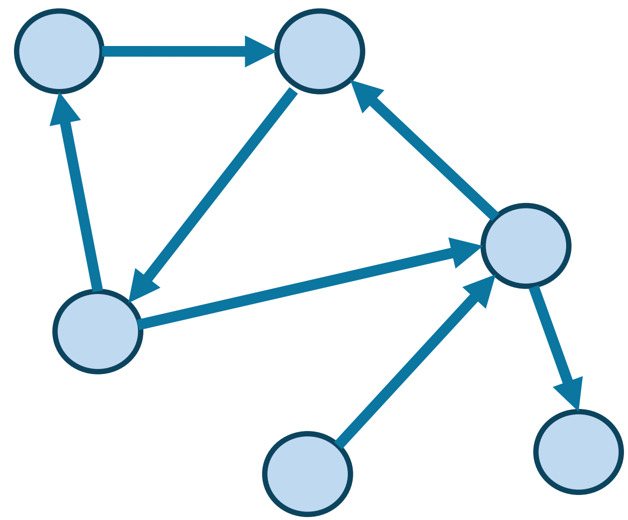
Planar Graph

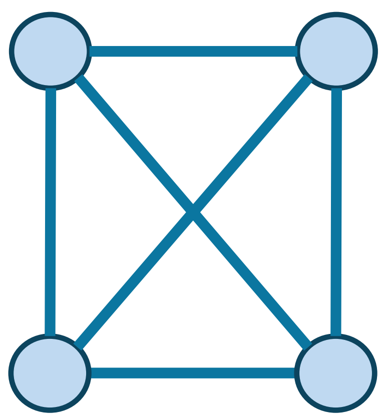
All edges can be drawn on a plane with none crossing



K3 is planar

This earlier graph is planar

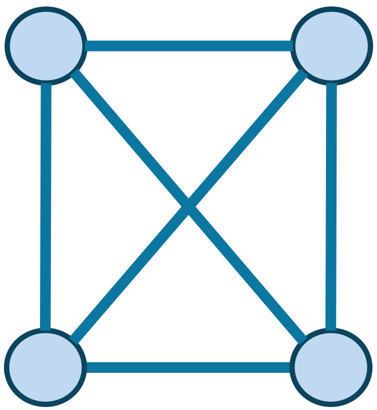




K4 planar?

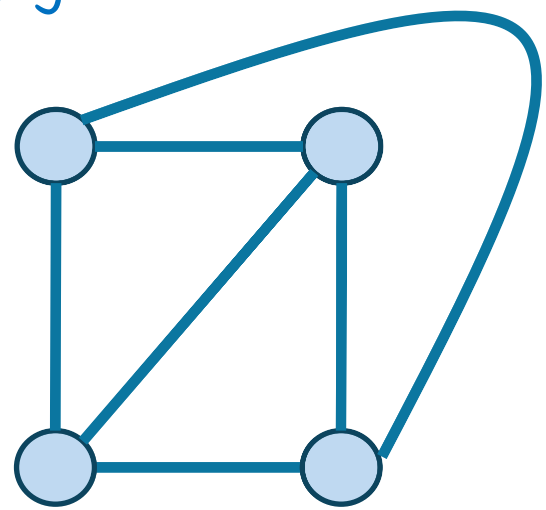
Planar Graph

May be planar but drawn poorly



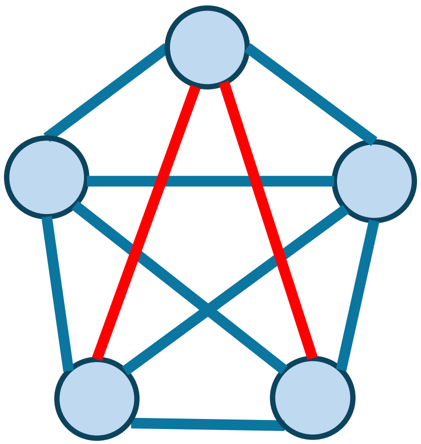
K4

Redraw one of the crossing edges, "rubber-band" it outside the others



K4 is planar

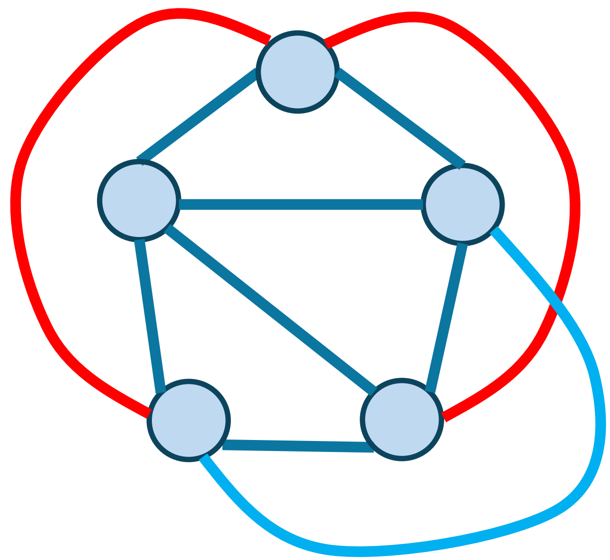
Planar Graph

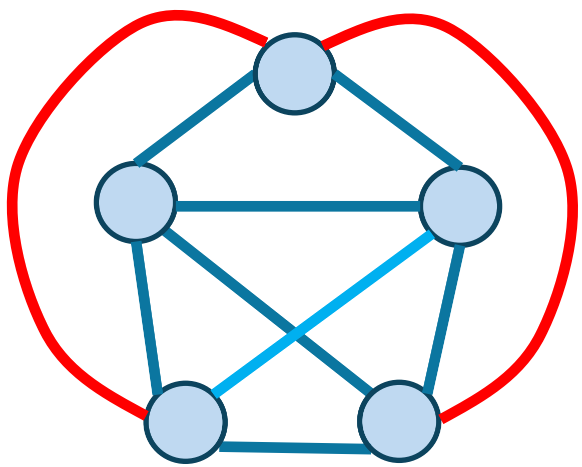


K5

not planar

What about K5?



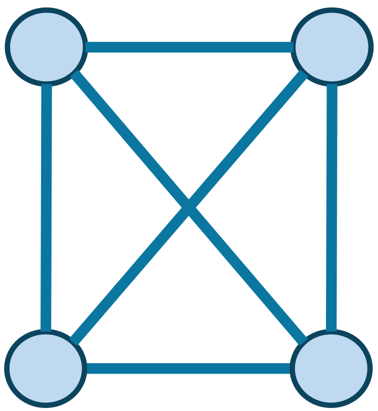


No matter which inner edge you "stretch" you get a cross

Turns out that K4 is the largest complete graph that is planar

Planar Graph

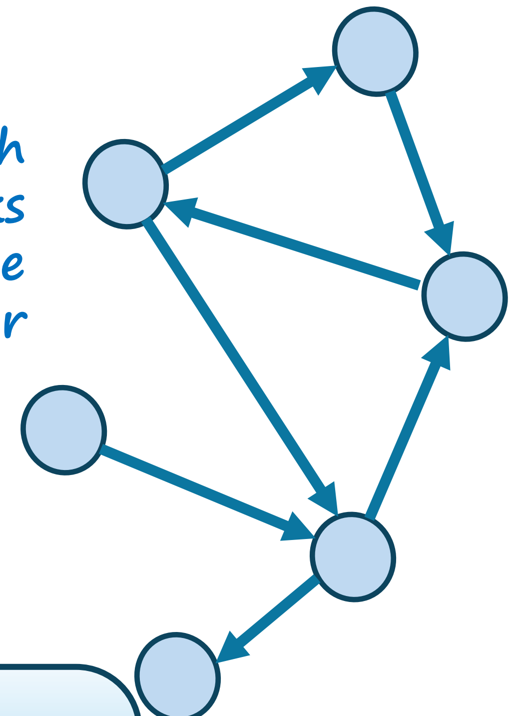
Rule of thumb



This graph "drawing" looks not planar...

The graph might still be planar... it just might be drawn poorly to show that

This graph "drawing" looks planar... so the graph IS planar



A graph is not the drawing... a graph is the math object

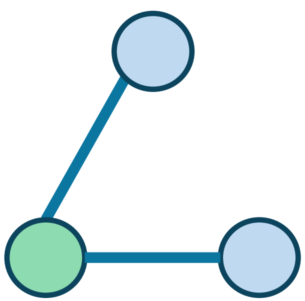
Bipartite Graph

Nodes are in two disjoint sets (types), and every edge connects different type nodes

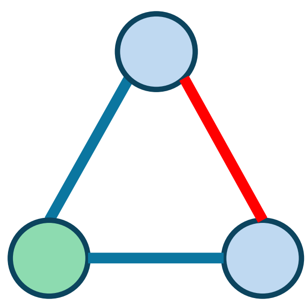
$V = V_1 \cup V_2$ *and*

$E = \{ (a, b) \mid (a \in V_1 \land b \in V_2) \lor (a \in V_2 \land b \in V_1) \}$

or, $E = \{ \{a, b\} \mid (a \in V_1 \land b \in V_2) \lor (a \in V_2 \land b \in V_1) \}$



is bipartite

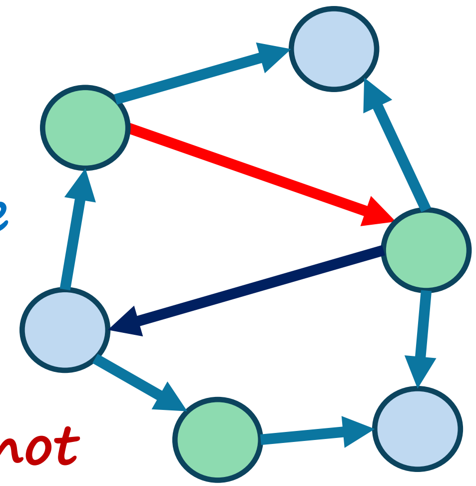


not bipartite

is bipartite

still is ...

but now not



More Bipartite

- Can think of bipartite as colorable with 2 colors
- Every edge goes between the 2 color collections