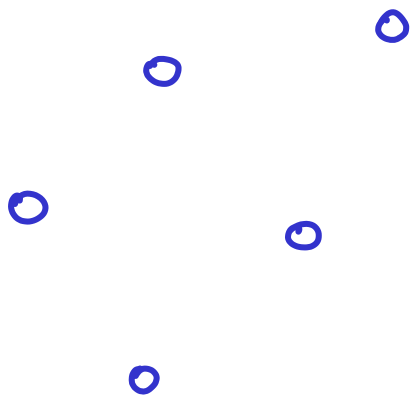
GRAPHS

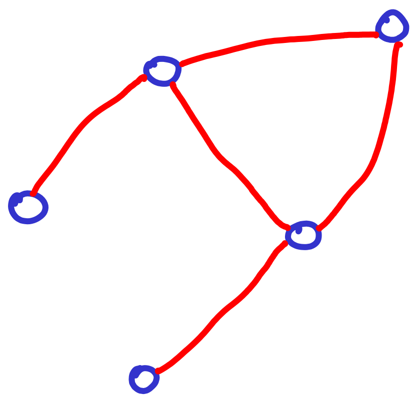
GRAPHS

vertices



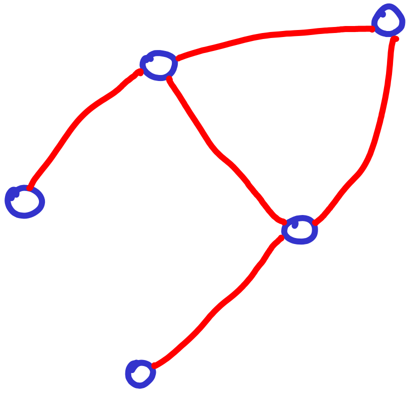
GRAPHS

vertices and edges



GRAPHS

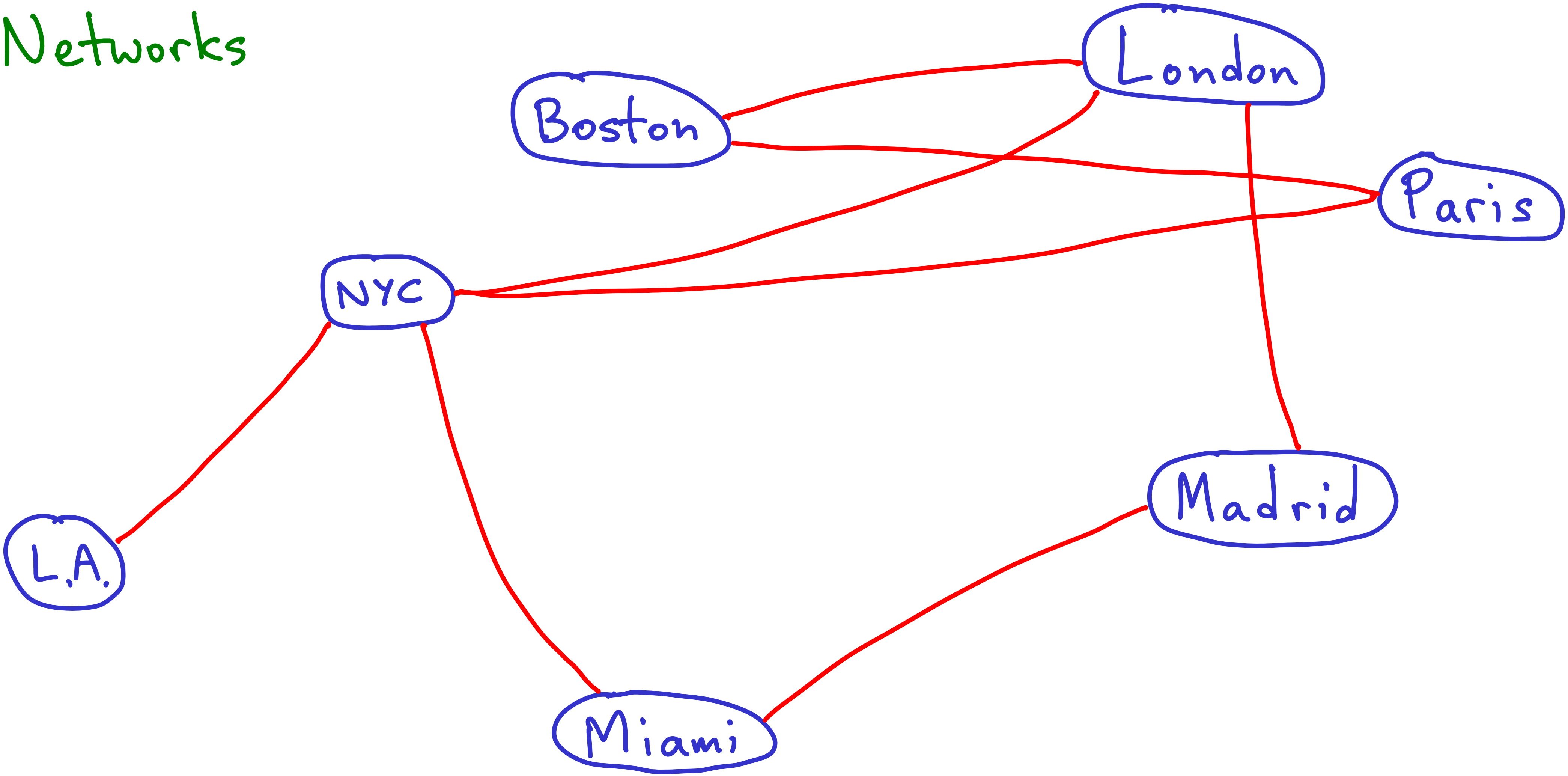
vertices and edges



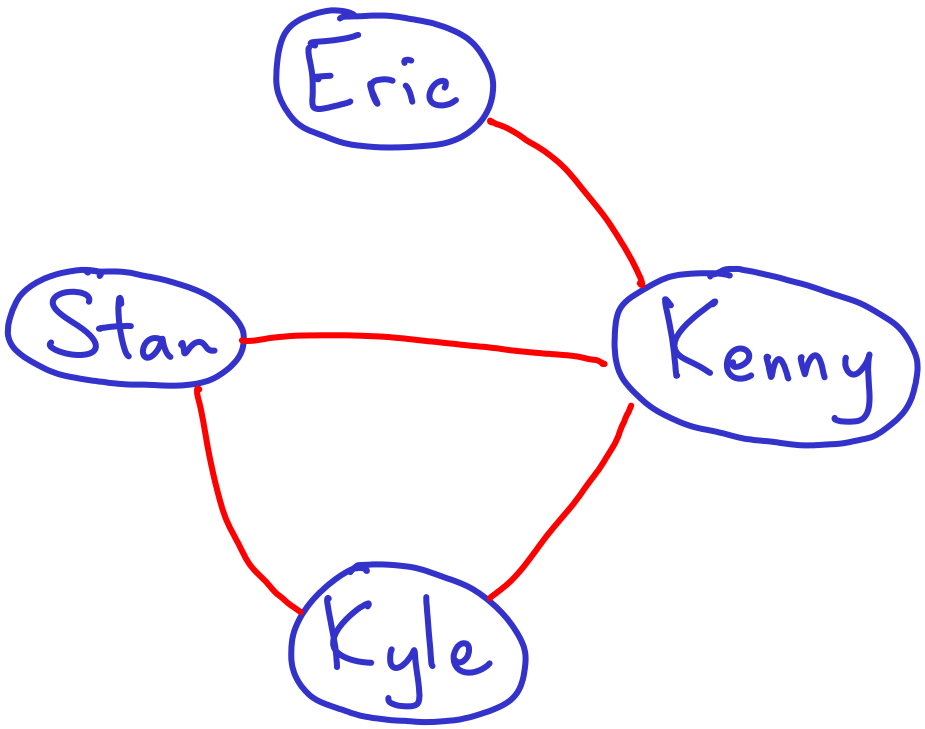
Terminology: one vertex • not "one vertice"

many vertices

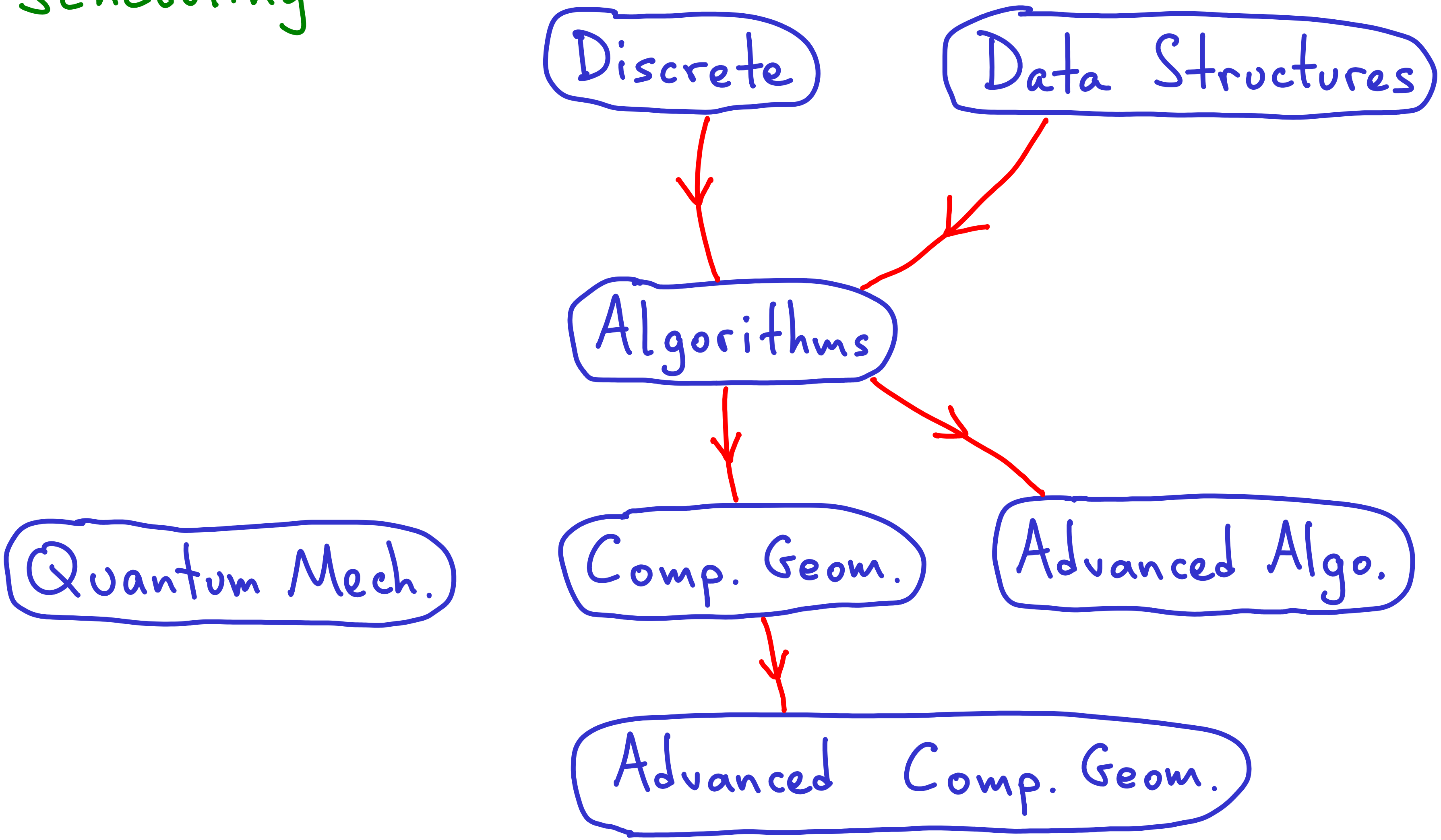
Networks



Social networks



Scheduling



(Directed Graph)

Graphs can be "abstract"

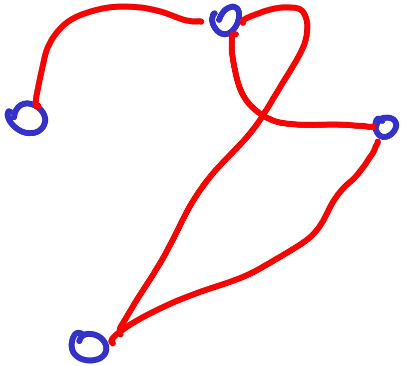
co-ordinates & drawings don't matter

Eric, Kenny
Kenny, Stan
Kenny, Kyle
Stan, Kyle

Graphs can be "abstract"

co-ordinates & drawings don't matter

although sometimes it helps to draw & visualize



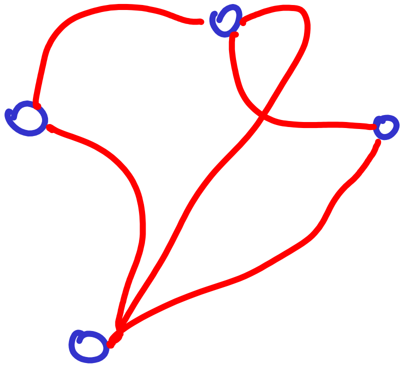
Eric, Kenny
Kenny, Stan
Kenny, Kyle
Stan, Kyle

Graphs can be "abstract" or "geometric" / "embedded"

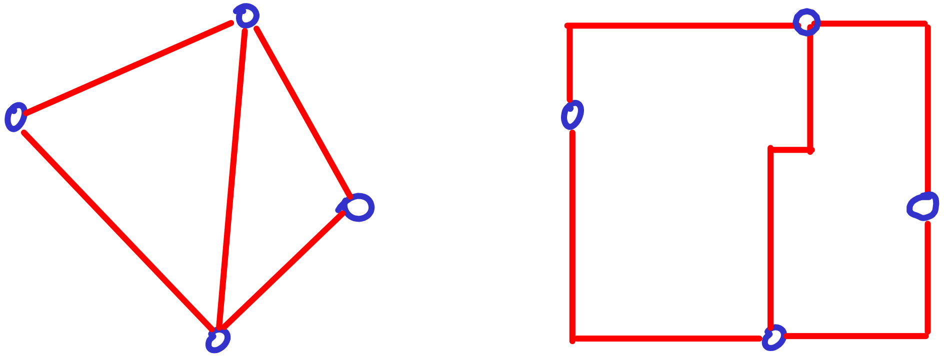
co-ordinates & drawings don't matter

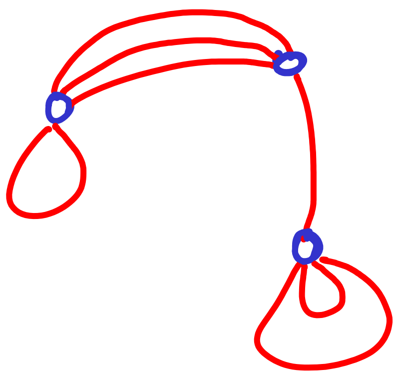
representing physical restrictions

although sometimes it helps to draw & visualize



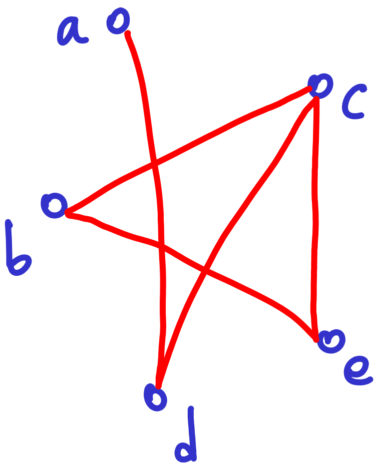
vs





Often it is assumed that there are no self-loops or multi-edges (duplicates)

Assume this unless specified



$G = (V, E)$

$V \downarrow V(G)$ $\quad E \downarrow E(G)$

V : set of vertices

E : set of edges

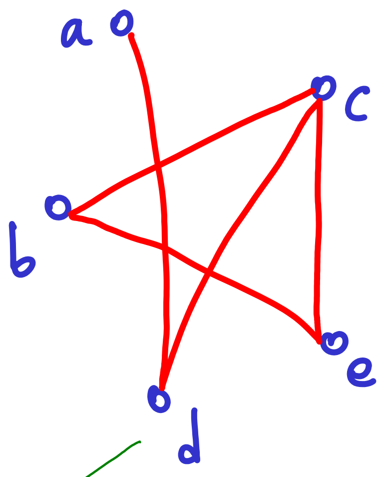
$(\{a, b, c, d, e\}, \{ad, bc, be, ce, cd\})$

$$G = (V, E)$$

$V \downarrow V(G)$ $E \downarrow E(G)$

V : set of vertices

E : set of edges



$$(\{a, b, c, d, e\}, \{ad, bc, be, ce, cd\})$$

Terminology:

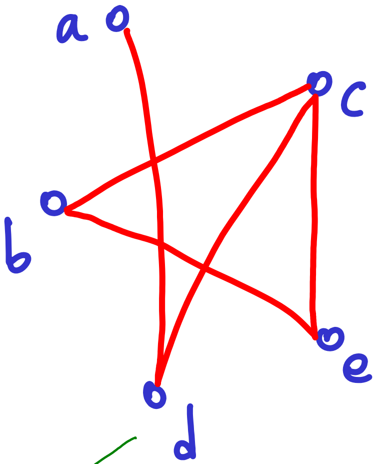
d is incident to two edges

$$G = (V, E)$$

$$V \to V(G) \qquad E \to E(G)$$

V : set of vertices

E : set of edges



$$(\{a, b, c, d, e\}, \{ad, bc, be, ce, cd\})$$

Terminology:

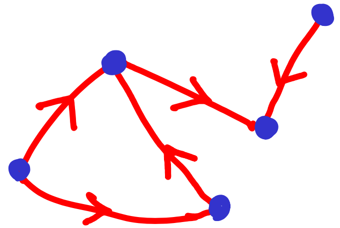
d is incident to two edges

Terminology:

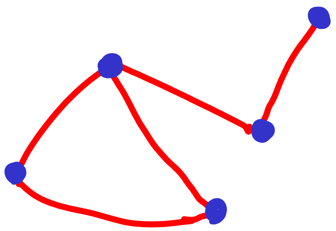
b & c
- are adjacent
- share an edge
- are neighbors

Some graph types

directed

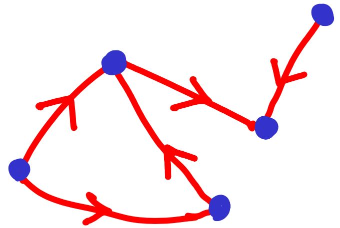


not directed

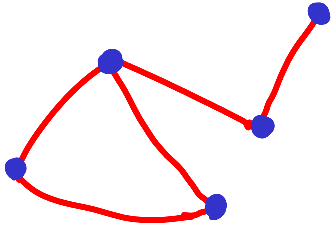


Some graph types

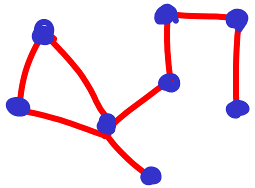
directed



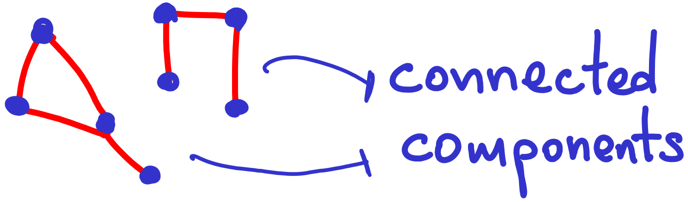
not directed



connected

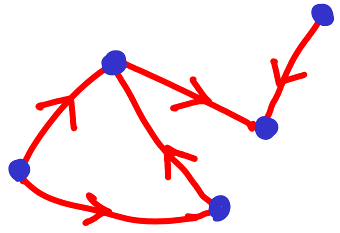


not connected

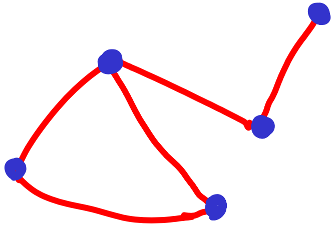


Some graph types

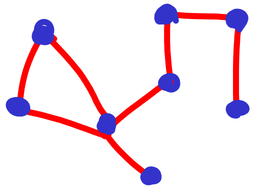
directed



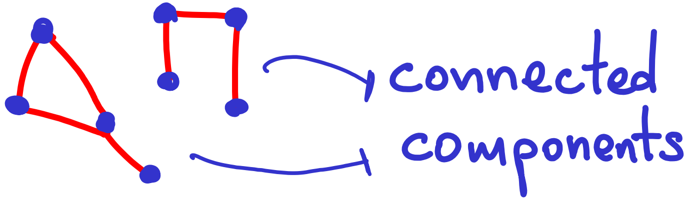
not directed



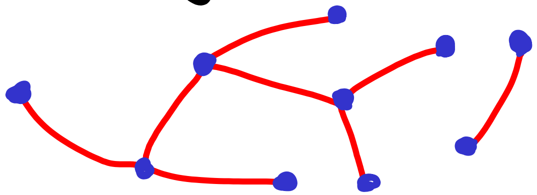
connected



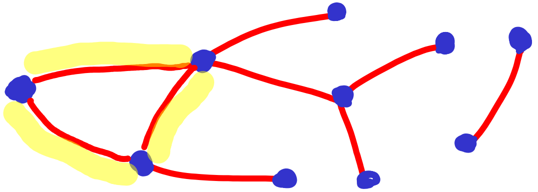
not connected



acyclic

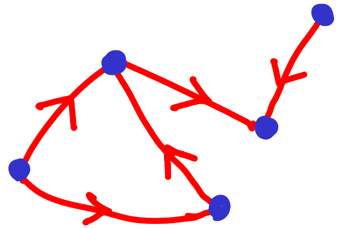


not acyclic

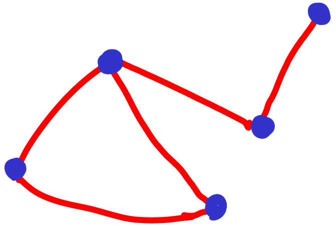


Some graph types

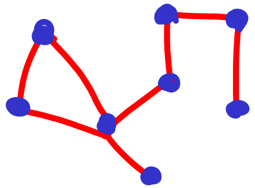
directed



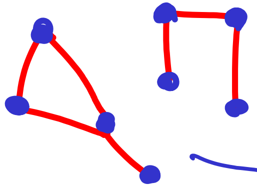
not directed



connected

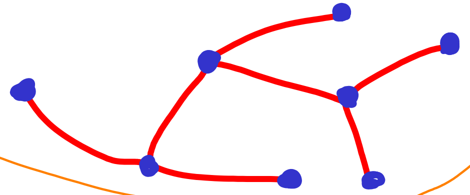


not connected

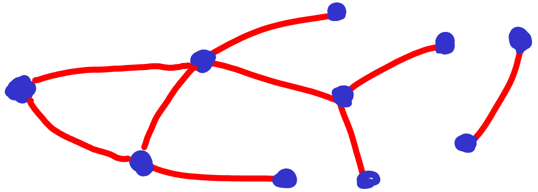


connected components

A connected acyclic graph is a tree

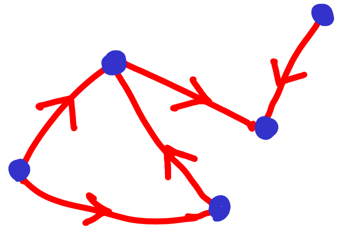


not acyclic

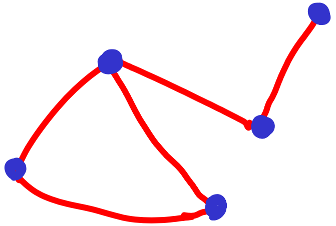


Some graph types

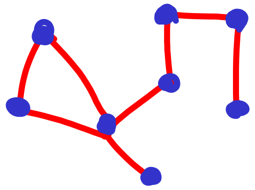
directed



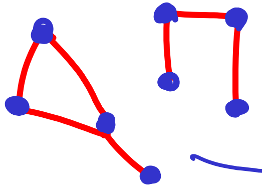
not directed



connected



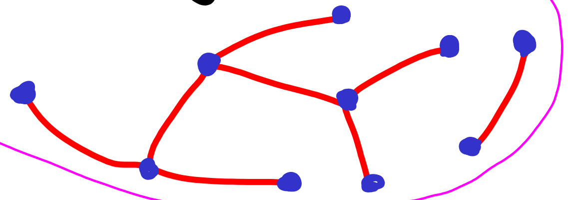
not connected



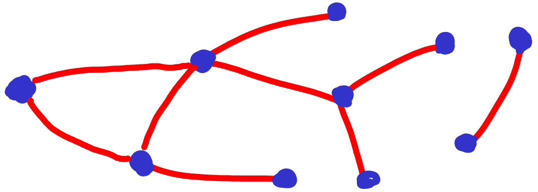
connected components

A connected acyclic graph is a tree

acyclic



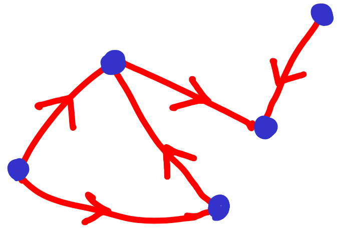
not acyclic



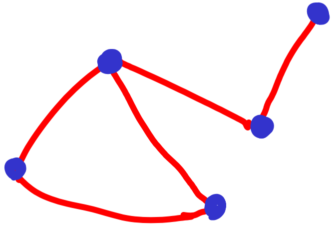
Forest of 2 trees

Some graph types

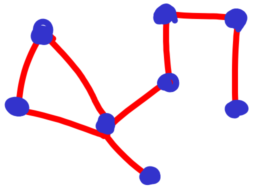
directed



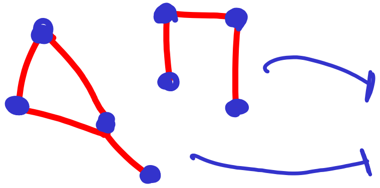
not directed



connected



not connected

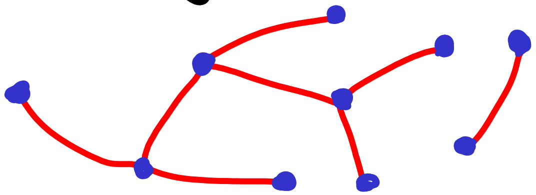


connected components

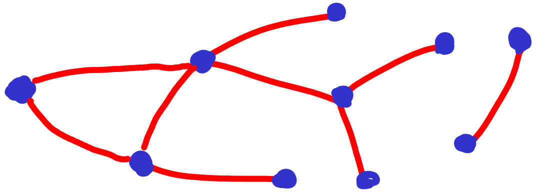
A connected acyclic graph is a tree

acyclic

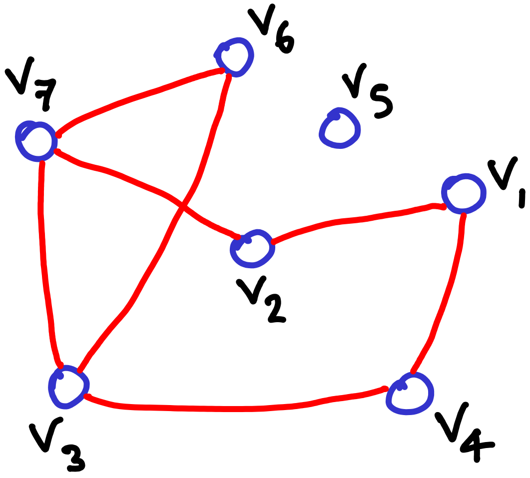
Forest of 2 trees →



not acyclic



Representation

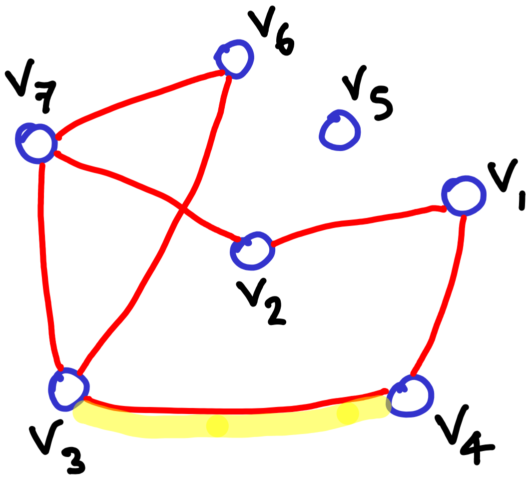


$$G = \{V, E\}$$

How can we store graphs without drawings

?

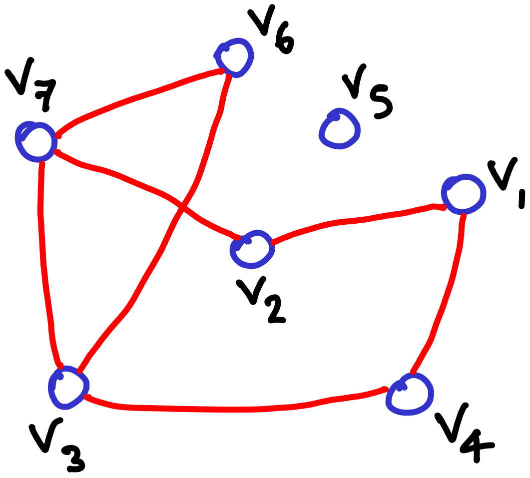
Representation



$$\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
3 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
7 & 0 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}\Bigg\}\ \text{Adjacency matrix}$$

$$G = \{V, E\}$$

Representation



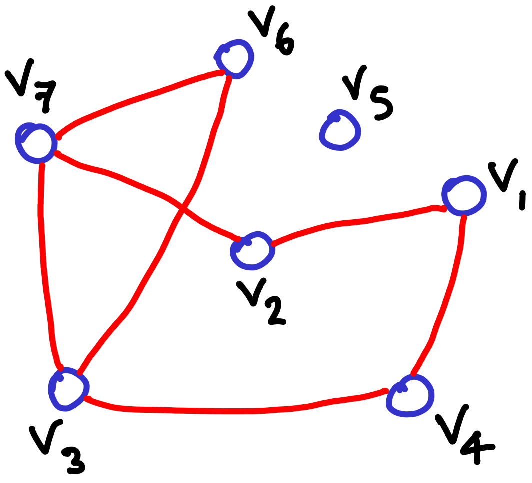
$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
3 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
7 & 0 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}
\Bigg\}
$$

Adjacency matrix

(symmetric for undirected)

$$G = \{V, E\}$$

Representation



$$G = \{V, E\}$$

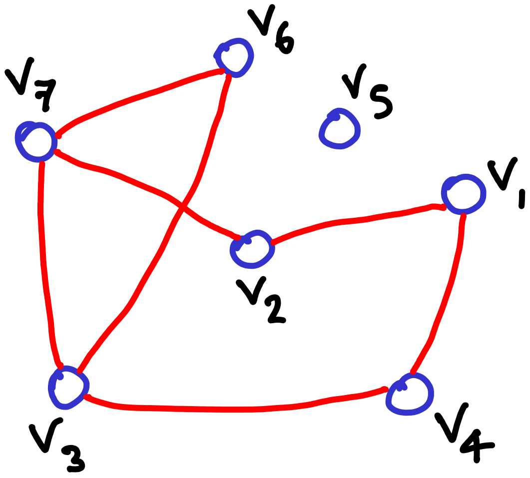
$$\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 7 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}$$

Adjacency matrix

size: ?

(symmetric for undirected)

Representation



$$G = \{V, E\}$$

$$\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
3 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
7 & 0 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}$$

Adjacency matrix

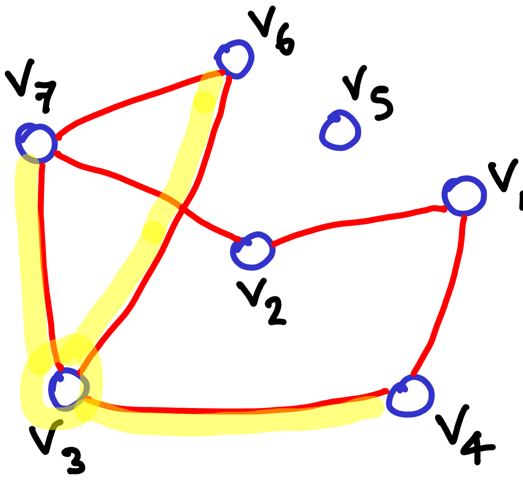
size: $|V|^2$

(symmetric for undirected)

$|V|$: size of vertex set

↳ often just written as V

Representation



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

Adjacency matrix

size: $|V|^2$

(symmetric for undirected)

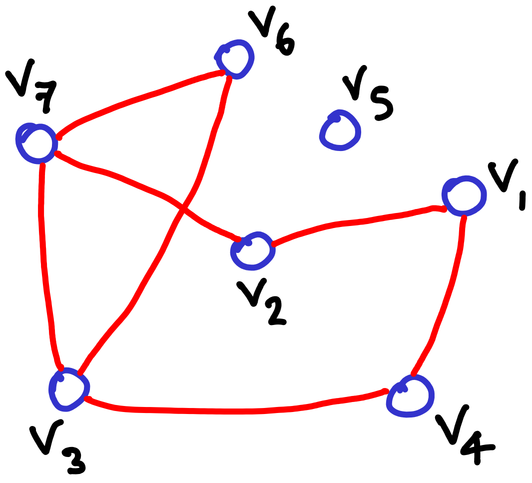
$G = \{V, E\}$

Adjacency list

1 → 4 → 2
2 → 1 → 7
3 → 4 → 7 → 6
4 → 1 → 3
5
6 → 3 → 7
7 → 6 → 3 → 2

For every vertex y in primary column (array) store a list of neighbors of y.

Representation



$G = \{V, E\}$

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

Adjacency matrix

size: $|V|^2$

(symmetric for undirected)

Adjacency list

size: ?

$1 \to 4 \to 2$

$2 \to 1 \to 7$

$3 \to 4 \to 7 \to 6$

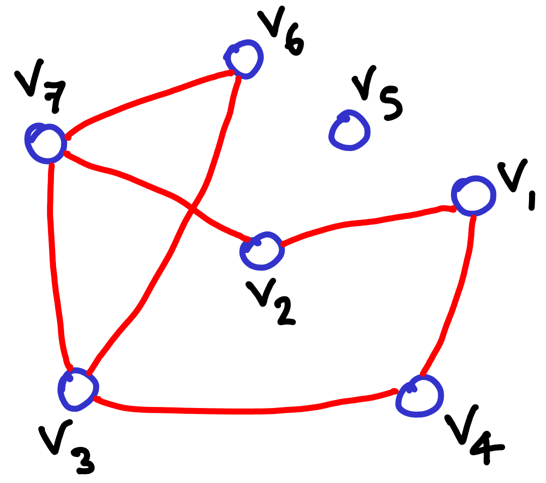
$4 \to 1 \to 3$

5

$6 \to 3 \to 7$

$7 \to 6 \to 3 \to 2$

Representation



$G = \{V, E\}$

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

Adjacency matrix

size: $|V|^2$

(symmetric for undirected)

Adjacency list

size: $|V| + 2|E|$

(undirected)

$1 \rightarrow 4 \rightarrow 2$

$2 \rightarrow 1 \rightarrow 7$

$3 \rightarrow 4 \rightarrow 7 \rightarrow 6$

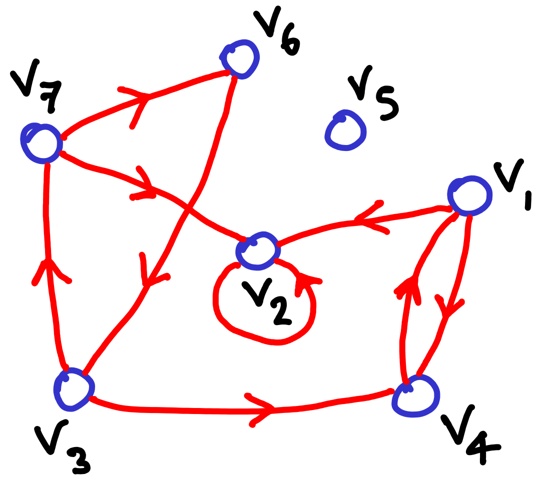
$4 \rightarrow 1 \rightarrow 3$

5

$6 \rightarrow 3 \rightarrow 7$

$7 \rightarrow 6 \rightarrow 3 \rightarrow 2$

Representation



$G = \{V, E\}$

directed

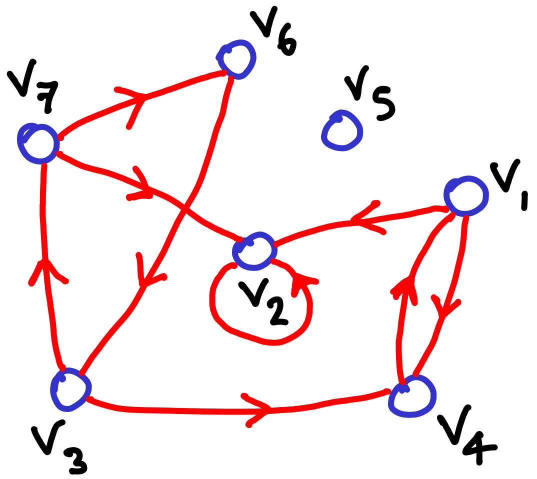
Adjacency matrix

size: ?

Adjacency list

size: ?

Representation



$$G = \{V, E\}$$

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

Adjacency matrix

size: $|V|^2$

(directed or not)

Adjacency list

size: $|V| + |E|$

(directed)

$1 \to 2 \to 4$
$2 \to 2$
$3 \to 4 \to 7$
$4 \to 1$
5
$6 \to 3$
$7 \to 6 \to 2$

$$\left.\begin{array}{ll} \text{Adjacency matrix} & \text{size: } \Theta(V^2) \\ \text{Adjacency list} & \text{size: } \Theta(V+E) \end{array}\right\} \text{ directed or not}$$

Adjacency matrix size: $\Theta(V^2)$

Adjacency list size: $\Theta(V+E)$

$\Big\}$ directed or not

Same for "dense" graphs, i.e. $E = \Theta(V^2)$

Adj. list uses less space for "sparse" graphs

Adjacency matrix size: $\Theta(V^2)$

Adjacency list size: $\Theta(V+E)$

} directed or not

Same for "dense" graphs, i.e. $E = \Theta(V^2)$

Adj. list uses less space for "sparse" graphs

- How much time to...
  - Query adjacency : Matrix ?

(are v_i & v_j neighbors?) List ?

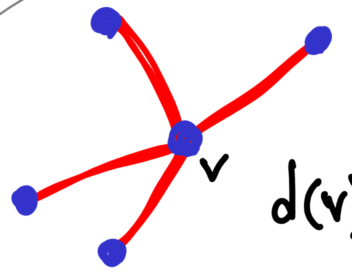
Adjacency matrix size: $\Theta(V^2)$

Adjacency list size: $\Theta(V+E)$

$\Big\}$ directed or not

Same for "dense" graphs, i.e. $E = \Theta(V^2)$

Adj. list uses less space for "sparse" graphs



$d(v) = 4$

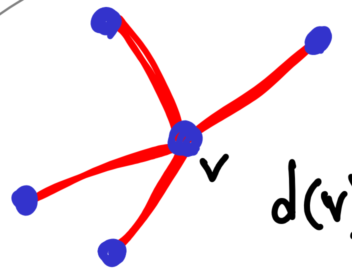
- How much time to...
  - Query adjacency: Matrix $\Theta(1)$

(are v_i & v_j neighbors?) List $O(V)$ ... but really $O(\text{degree}(v_i))$

or $d(v_j)$

Adjacency matrix size: $\Theta(V^2)$
Adjacency list size: $\Theta(V+E)$
} directed or not

Same for "dense" graphs, i.e. $E = \Theta(V^2)$
Adj. list uses less space for "sparse" graphs



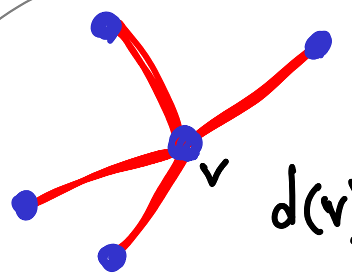
$d(v)=4$

- How much time to...
  - Query adjacency: Matrix $\Theta(1)$
    (are v_i & v_j neighbors?) List $O(V)$ ... but really $O(\text{degree}(v_i))$ or $d(v_j)$
  - Enumerate neighbors: List ?
    (of one vertex, v_k) Matrix ?

Adjacency matrix size: $\Theta(V^2)$ }
Adjacency list size: $\Theta(V+E)$ } directed or not

Same for "dense" graphs, i.e. $E = \Theta(V^2)$

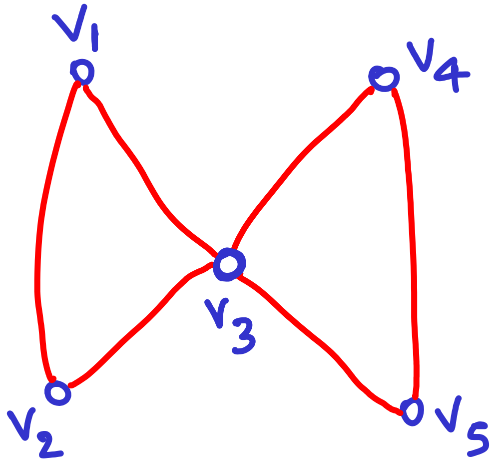
Adj. list uses less space for "sparse" graphs



$d(v)=4$

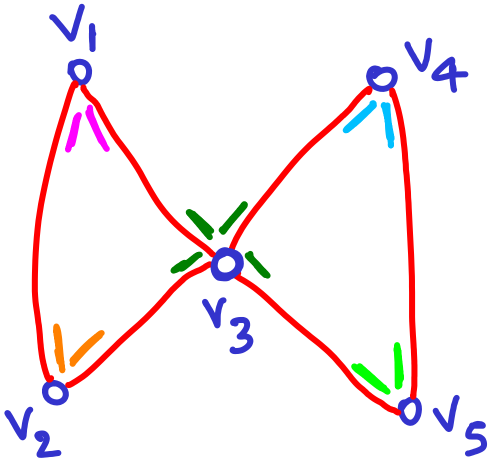
- How much time to...
  - Query adjacency: Matrix $\Theta(1)$
    (are v_i & v_j neighbors?) List $O(V)$ ... but really $O(\text{degree}(v_i))$ or $d(v_j)$
  - Enumerate neighbors: List $\Theta(d(v_k))$
    (of one vertex, v_k) Matrix $\Theta(V)$

Vertex degree



v_3 has 4 neighbors $\longleftrightarrow$ $d(v_3) = 4$

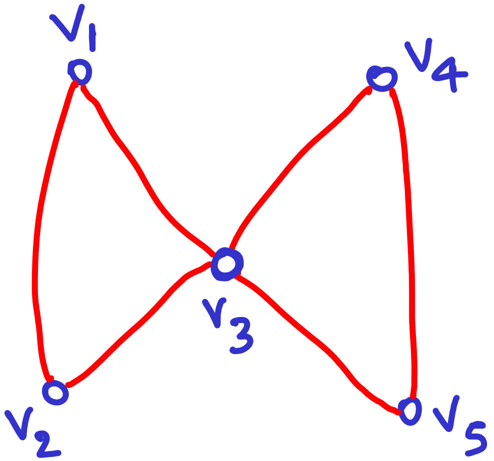
Vertex degree



v_3 has 4 neighbors $\longleftrightarrow$ $d(v_3) = 4$

$$\sum d(v) = 2+2+4+2+2 = 12$$

Vertex degree

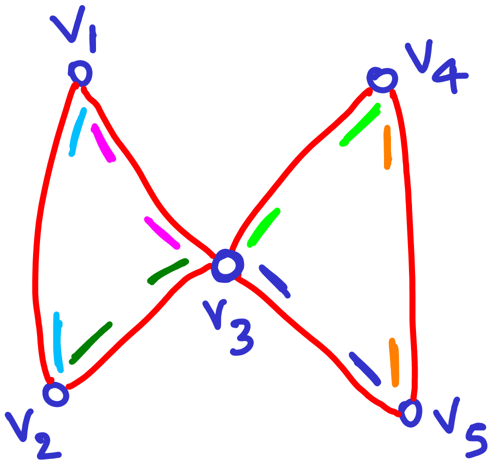


v_3 has 4 neighbors $\longleftrightarrow$ $d(v_3) = 4$

$$\sum d(v) = 2+2+4+2+2 = 12$$

$$|E| = 6$$

Vertex degree



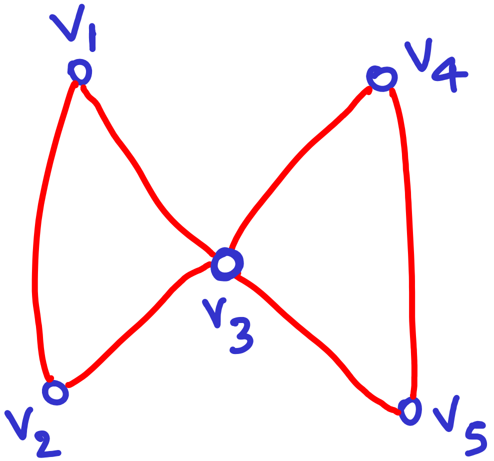
v_3 has 4 neighbors $\longleftrightarrow$ $d(v_3) = 4$

$\sum d(v) = 2+2+4+2+2 = 12$

$|E| = 6$

$2 \cdot |E|$ (doublecounting every edge)

Vertex degree



Handshaking theorem

v_3 has 4 neighbors $\longleftrightarrow$ $d(v_3) = 4$

$$\sum d(v) = 2+2+4+2+2 = 12$$

$$|E| = 6$$

$$\boxed{\sum_{v \in G} d(v) = 2 \cdot |E|}$$

(doublecounting every edge)

Regular graphs : all vertices have the same degree

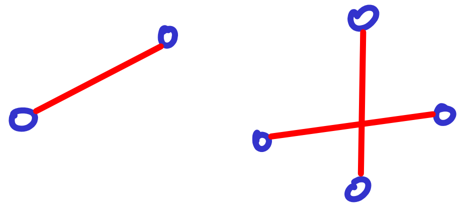
Regular graphs: all vertices have the same degree

$d=1$?

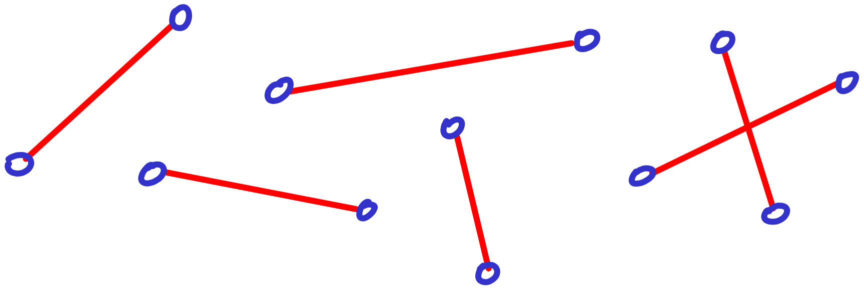
What graphs can you draw with this property?

Regular graphs: all vertices have the same degree

$d=1$?

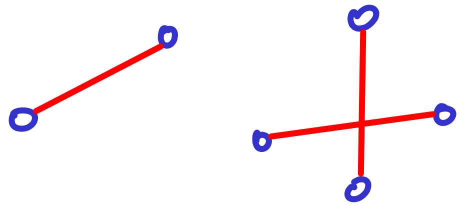


1-regular



Regular graphs : all vertices have the same degree

$d=1$?

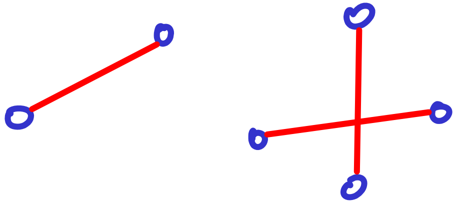


1-regular

$d=2$?

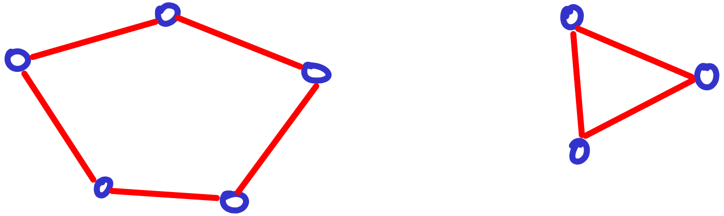
Regular graphs : all vertices have the same degree

$d=1$?



1-regular

$d=2$?

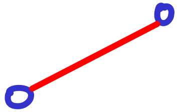


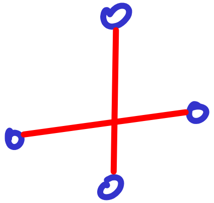
2-regular

collections of cycles

Regular graphs: all vertices have the same degree

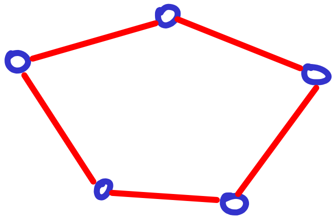
$d=1$?

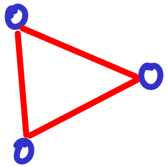




1-regular

$d=2$?



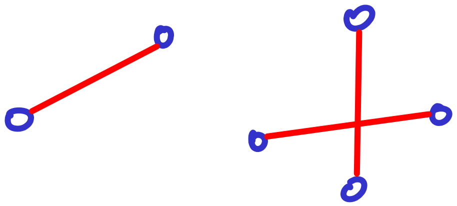


2-regular

$d=3$?

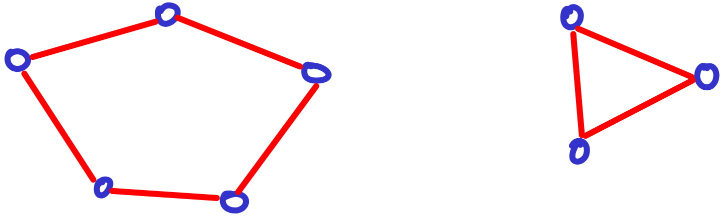
Regular graphs : all vertices have the same degree

d = 1 ?



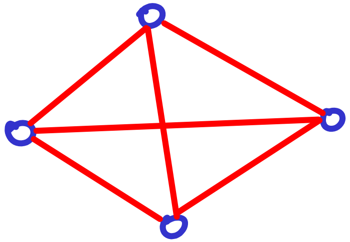
1-regular

d = 2 ?



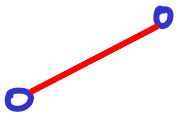
2-regular

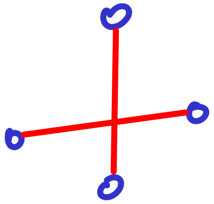
d = 3 ?



Regular graphs : all vertices have the same degree

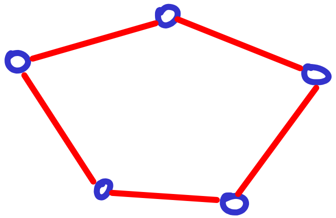
d = 1 ?

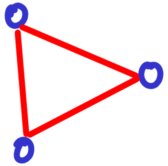




1-regular

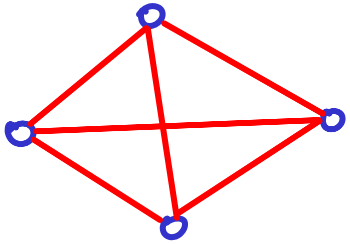
d = 2 ?

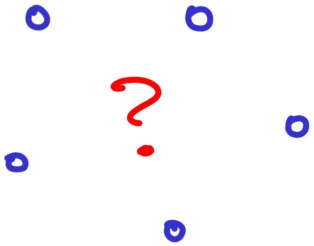




2-regular

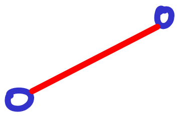
d = 3 ?

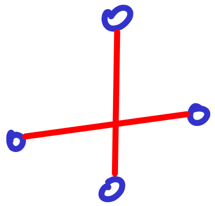




Regular graphs: all vertices have the same degree

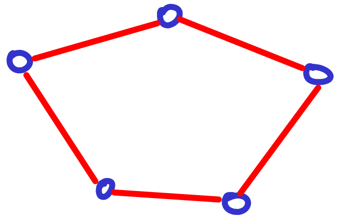
$d=1$?

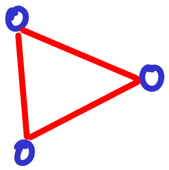




1-regular

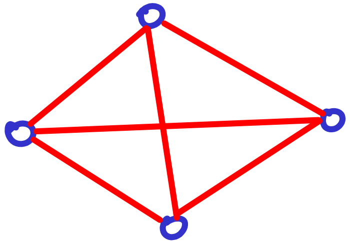
$d=2$?

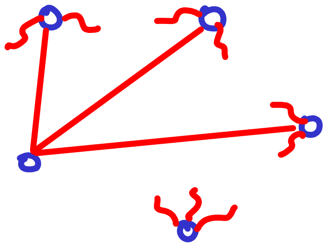




2-regular

$d=3$?

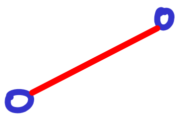


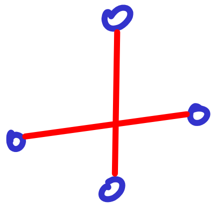


need $\sum d(v) = 15$

Regular graphs : all vertices have the same degree

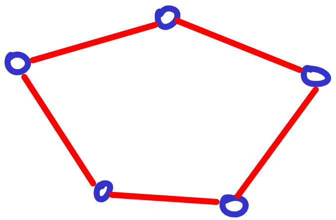
$d = 1$?

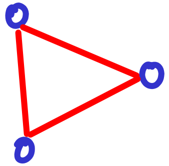




1-regular

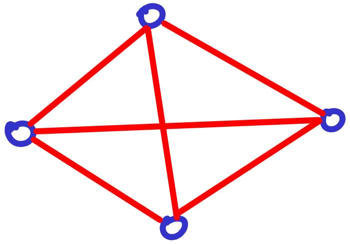
$d = 2$?

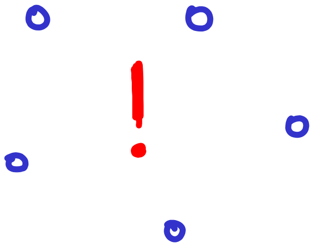




2-regular

$d = 3$?



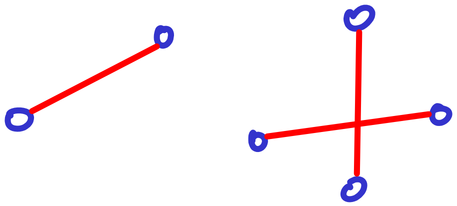


need $\sum d(v) = 15$

but $2E$ is even

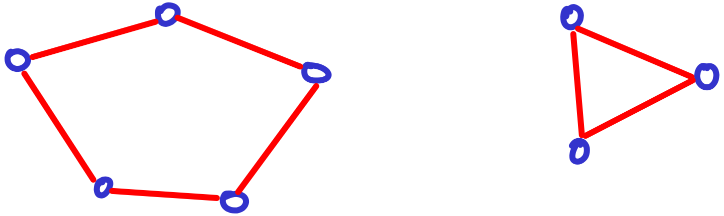
Regular graphs: all vertices have the same degree

$d=1$?



1-regular

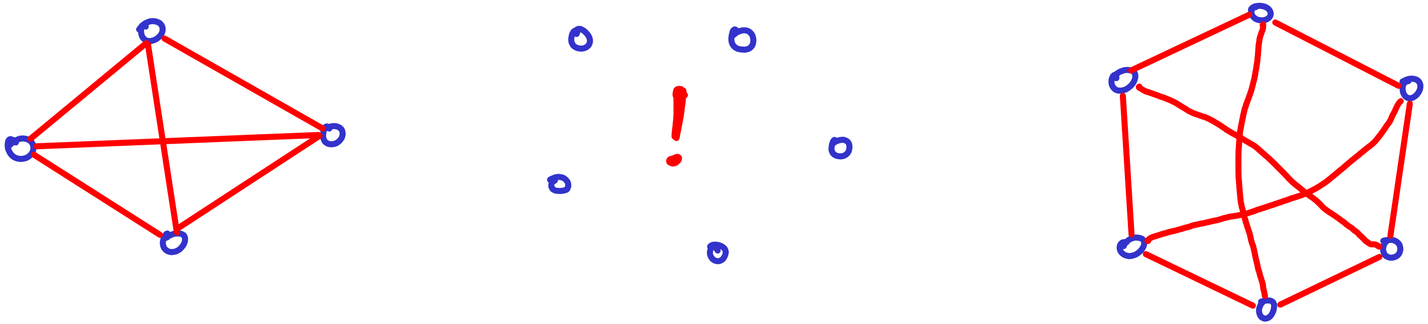
$d=2$?



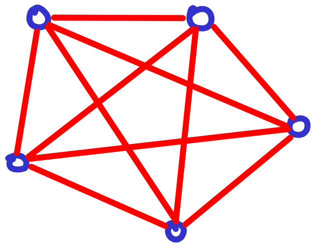
2-regular

3-regular

$d=3$?



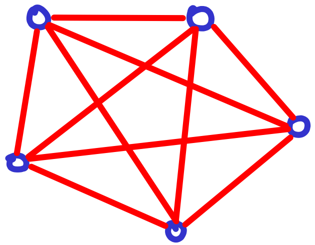
$V = 5$



4-regular

($V-1$ regular)

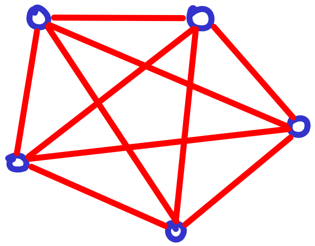
$V=5$



4-regular

($V-1$ regular) $\rightarrow$ complete graph : K_5

$V=5$



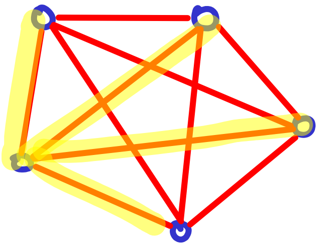
4-regular

(V-1 regular) $\rightarrow$ complete graph : K_5

#edges?

for K_5, and in general for K_V

$V = 5$



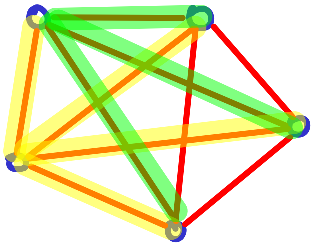
4-regular

($V-1$ regular) $\rightarrow$ complete graph : K_5

edges?

$\underline{V-1} + \ldots$

$V = 5$



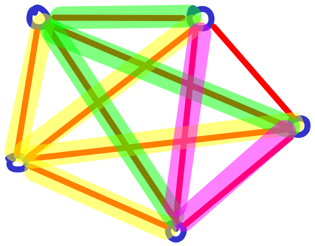
4-regular

(V-1 regular) → complete graph : K_5

edges?

$\underbrace{V-1}$ + $\underbrace{V-2}$ + ...

$V = 5$



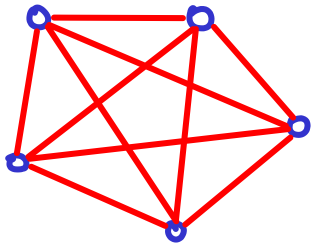
4-regular

($V-1$ regular) → complete graph : K_5

edges?

$\underline{V-1} + \underline{V-2} + \underline{V-3} + \dots$

$V=5$



4-regular

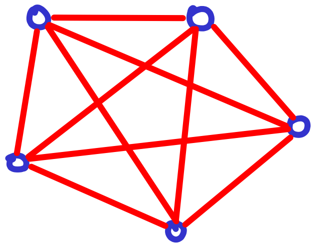
($V-1$ regular) $\rightarrow$ complete graph : K_5

#edges?

$$V-1 + V-2 + V-3 + \cdots + 3 + 2 + 1 = \binom{V}{2}$$

$$= \sum_{i=1}^{V-1} i = \frac{V \cdot (V-1)}{2}$$

$V=5$



4-regular

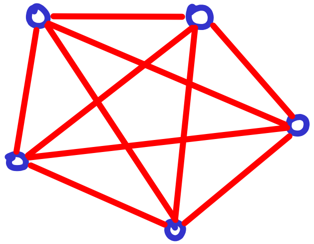
(V-1 regular) → complete graph : K_5

edges? $\quad V-1 + V-2 + V-3 + \cdots + 3 + 2 + 1 = \binom{V}{2}$

$$= \sum_{i=1}^{V-1} i = \frac{V\cdot(V-1)}{2}$$

Also: $\sum d(v) = \underbrace{V}_{\text{vertices}} \cdot \underbrace{(V-1)}_{\text{degree}}$

$V=5$



4-regular

$(V-1 \text{ regular}) \rightarrow$ complete graph : K_5

#edges? $V-1 + V-2 + V-3 + \cdots + 3 + 2 + 1 = \binom{V}{2}$

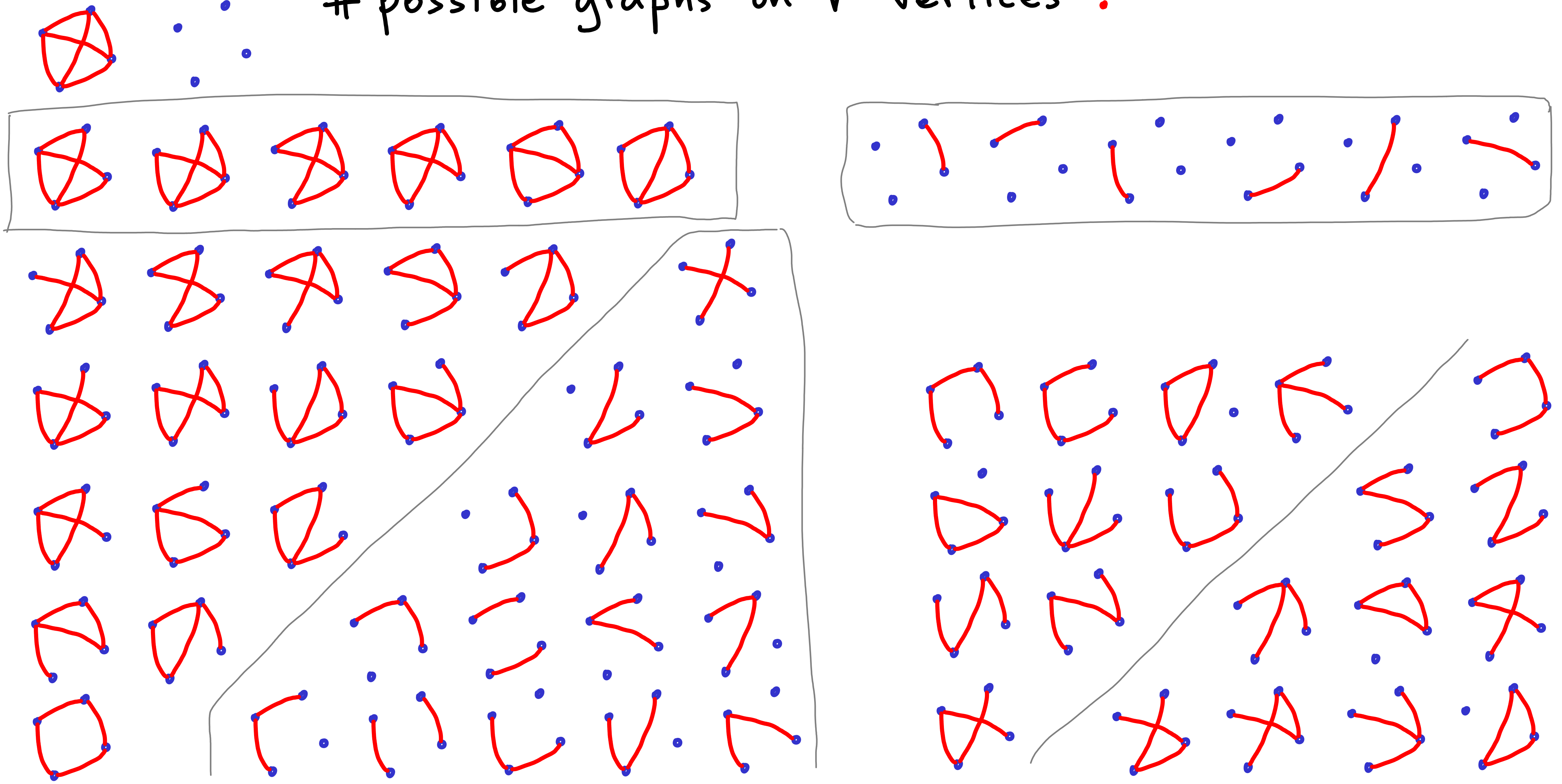
$$= \sum_{i=1}^{V-1} i = \frac{V\cdot(V-1)}{2}$$

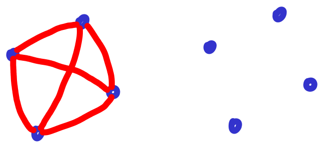
Also: $\sum d(v) = V\cdot(V-1)$

by Handshaking, $E = \dfrac{V\cdot(V-1)}{2}$

possible graphs on V vertices ?

possible graphs on V vertices ?

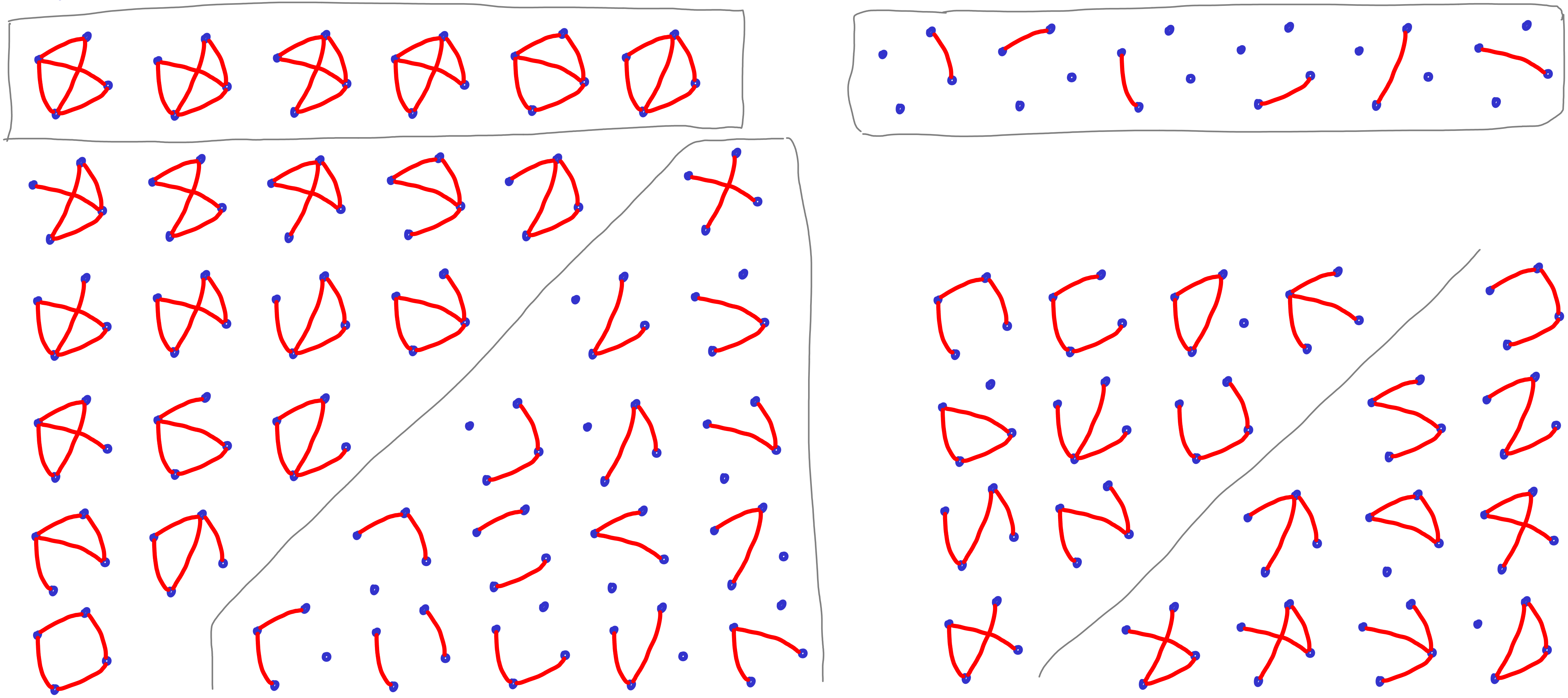




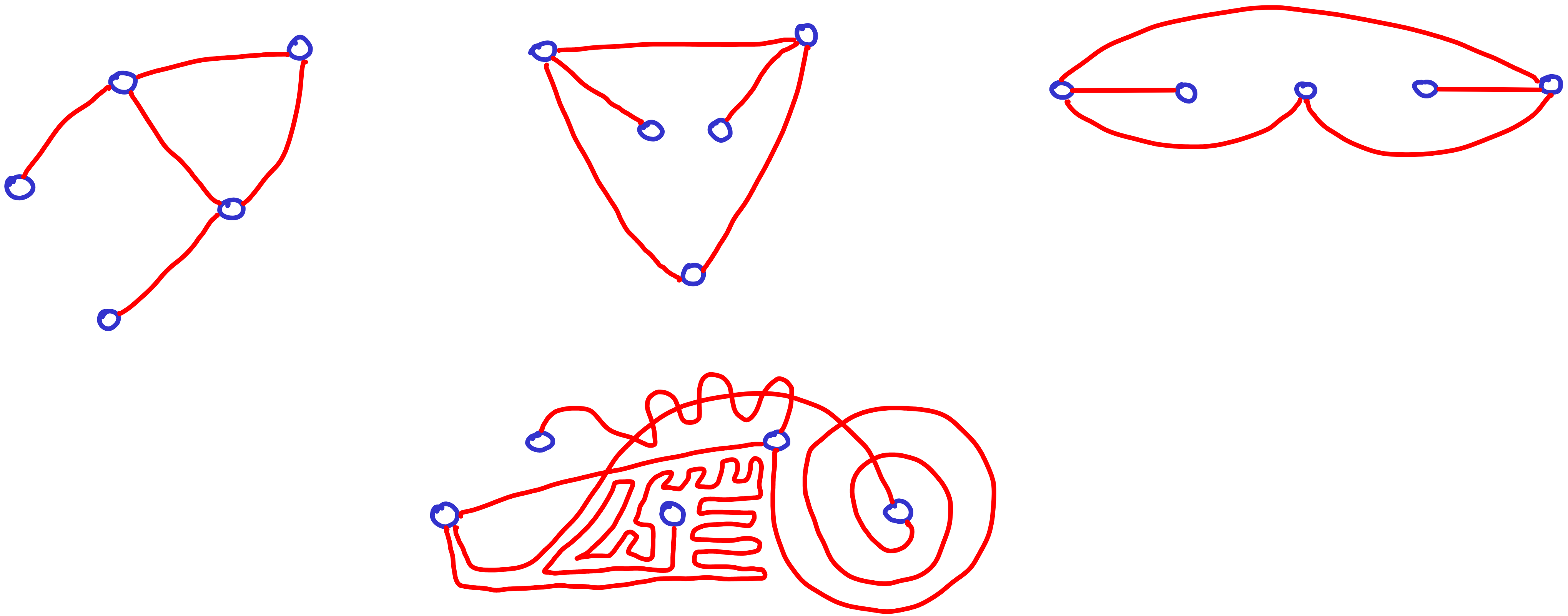
possible graphs on V vertices $= 2^{\binom{V}{2}}$ $\{0,1\}^{\binom{V}{2}}$

(each edge is IN or OUT)

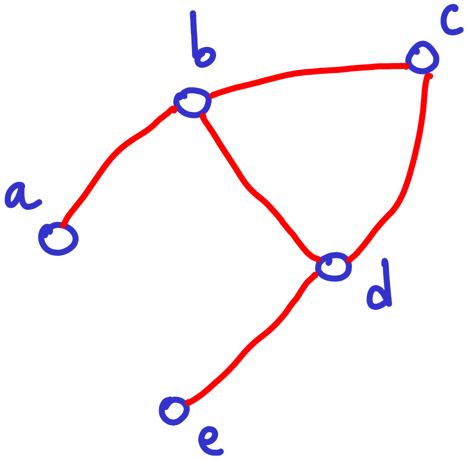
↳ but there are similar shapes

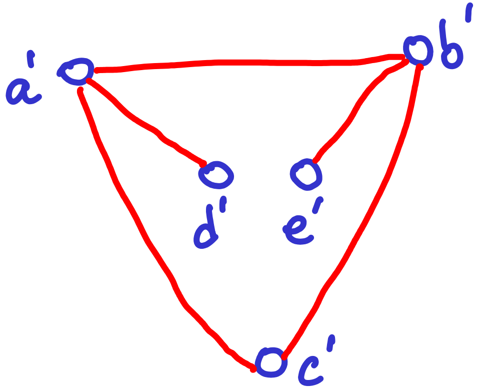


isomorphism



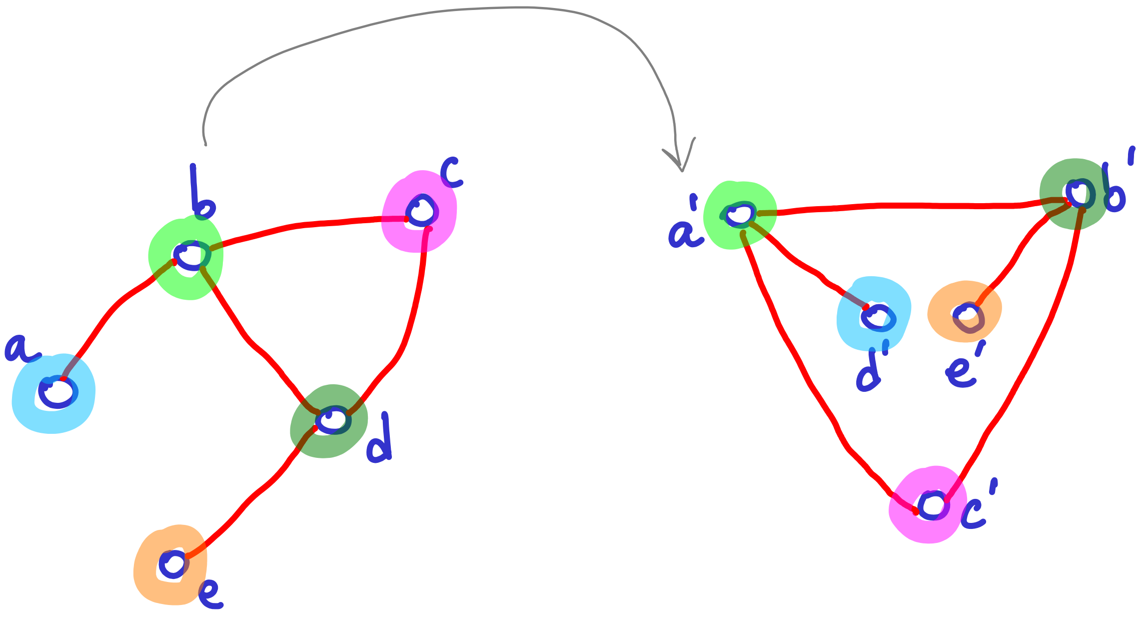
isomorphism $\rightarrow$ map vertices of graph G to vertices of graph H.





v	m(v)
a :	d'
b :	a'
c :	c'
d :	b'
e :	e'

isomorphism → map vertices of graph G to vertices of graph H.

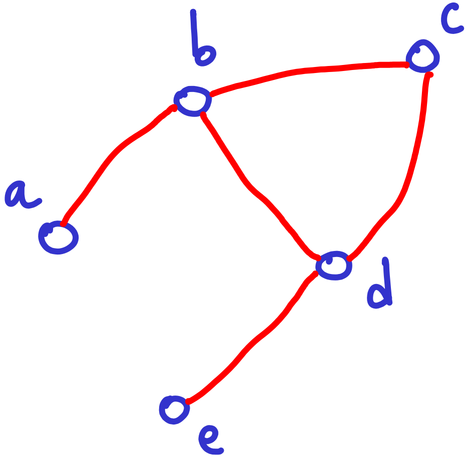


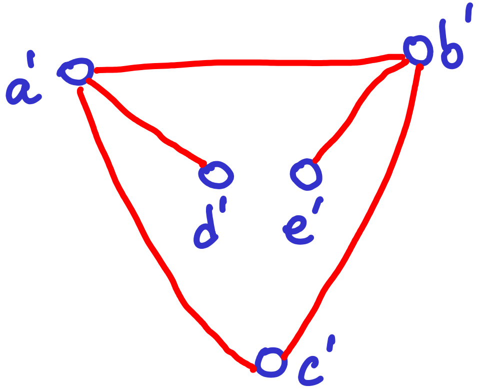
v	m(v)
a	d'
b	a'
c	c'
d	b'
e	e'

isomorphism → map vertices of graph G to vertices of graph H.

vertices $\alpha, \beta \in G$ share an edge in G

IFF vertices $m(\alpha), m(\beta) \in H$ share an edge in H.



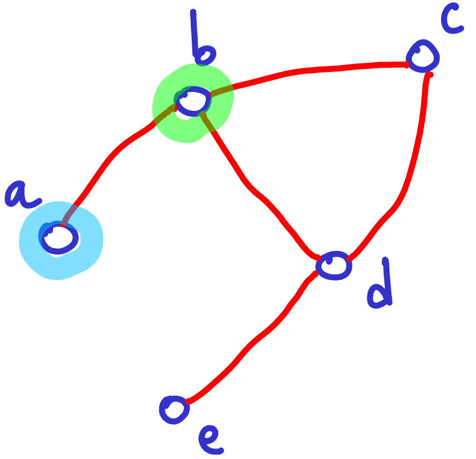


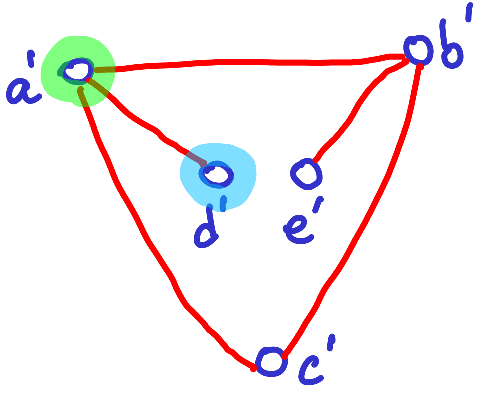
v	m(v)
a	: d'
b	: a'
c	: c'
d	: b'
e	: e'

isomorphism → map vertices of graph G to vertices of graph H.

vertices $\alpha, \beta \in G$ share an edge in G

IFF vertices $m(\alpha), m(\beta) \in H$ share an edge in H.



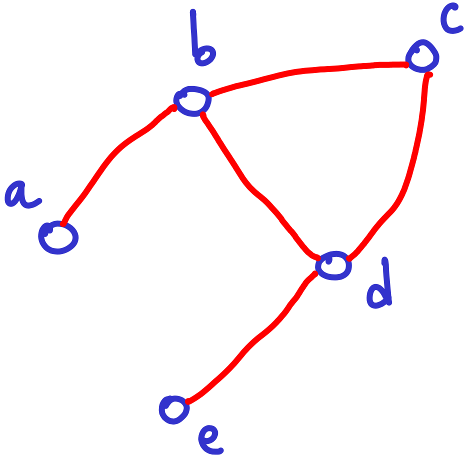


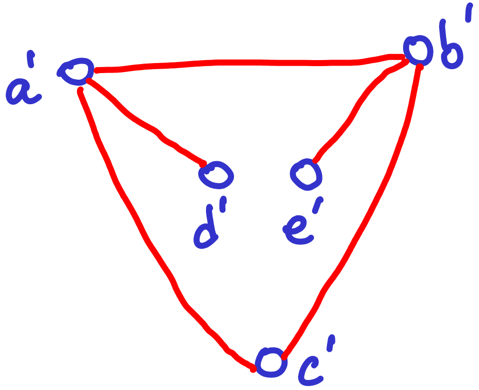
v	m(v)
a	: d'
b	: a'
c	: c'
d	: b'
e	: e'

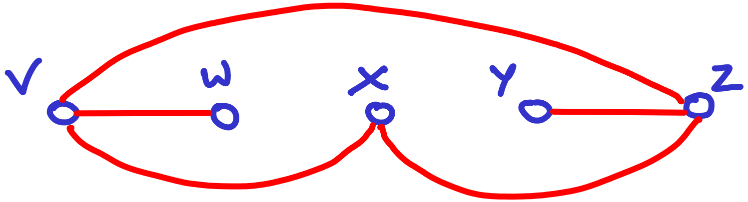
isomorphism → map vertices of graph G to vertices of graph H.

vertices $\alpha, \beta \in G$ share an edge in G

IFF vertices $m(\alpha), m(\beta) \in H$ share an edge in H.





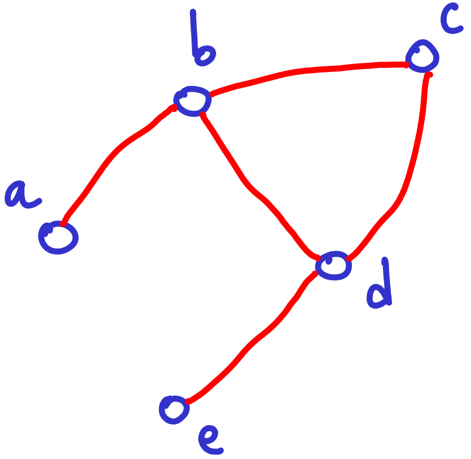


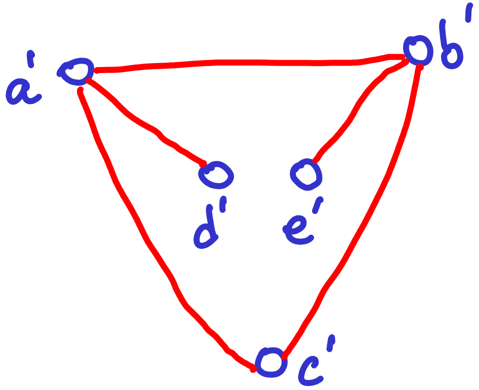
v		$m(v)$
a	: d'	: ?
b	: a'	: ?
c	: c'	: ?
d	: b'	: ?
e	: e'	: ?

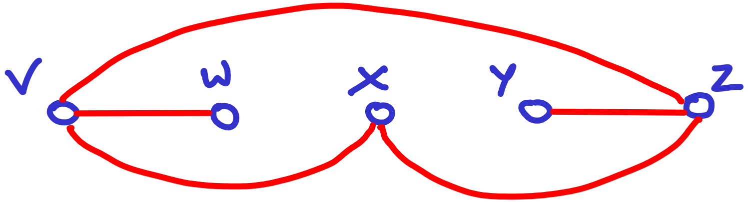
isomorphism → map vertices of graph G to vertices of graph H.

vertices $\alpha, \beta \in G$ share an edge in G

IFF vertices $m(\alpha), m(\beta) \in H$ share an edge in H.





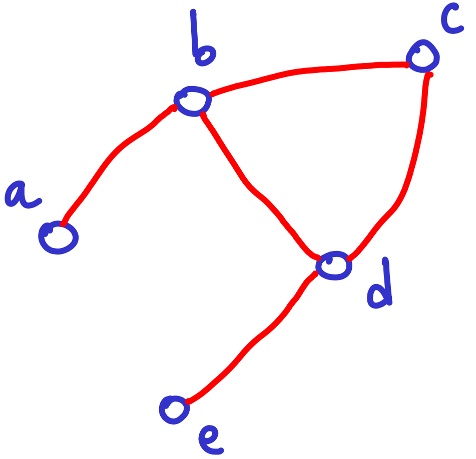


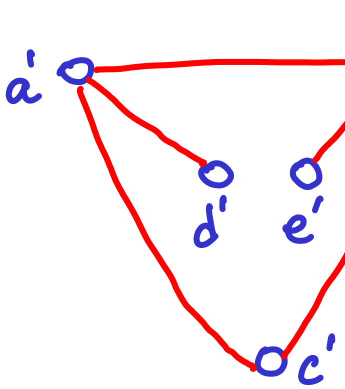
v		m(v)
a	d'	w
b	a'	v
c	c'	x
d	b'	z
e	e'	y

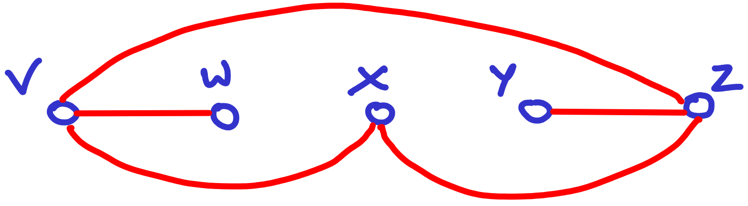
isomorphism → map vertices of graph G to vertices of graph H.

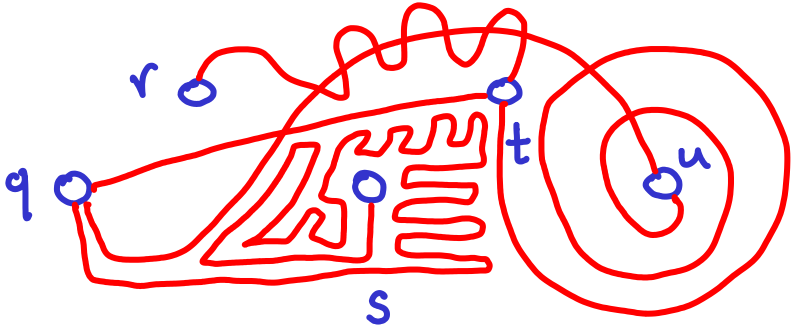
vertices $\alpha, \beta \in G$ share an edge in G

IFF vertices $m(\alpha), m(\beta) \in H$ share an edge in H.







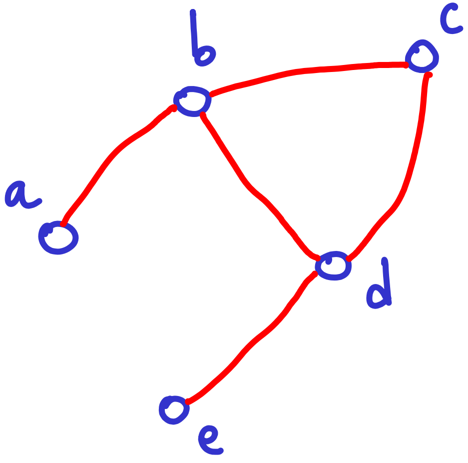


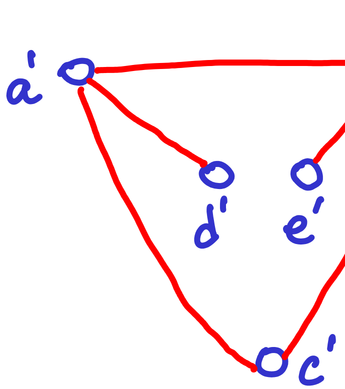
v			m(v)
a	d'	w	?
b	a'	v	?
c	c'	x	?
d	b'	z	?
e	e'	y	?

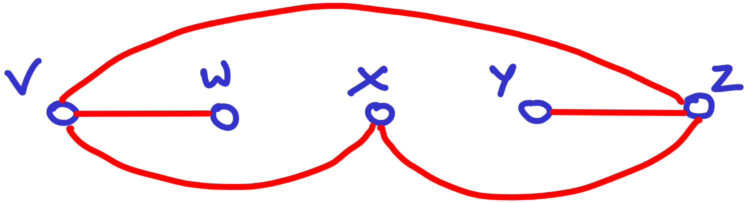
isomorphism → map vertices of graph G to vertices of graph H.

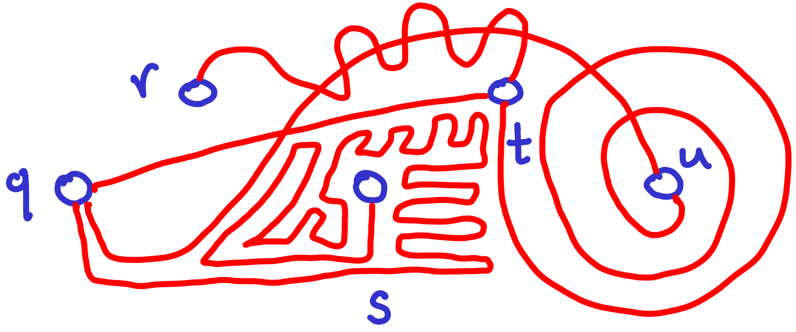
vertices $\alpha, \beta \in G$ share an edge in G

IFF vertices $m(\alpha), m(\beta) \in H$ share an edge in H.







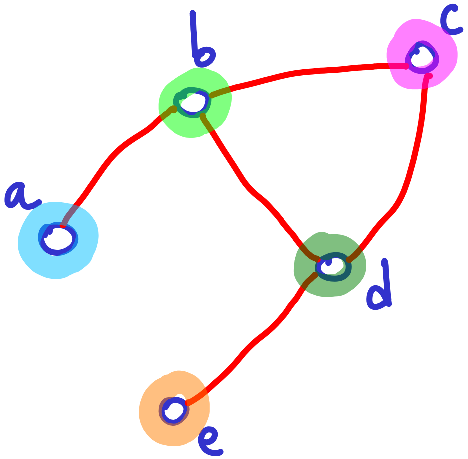


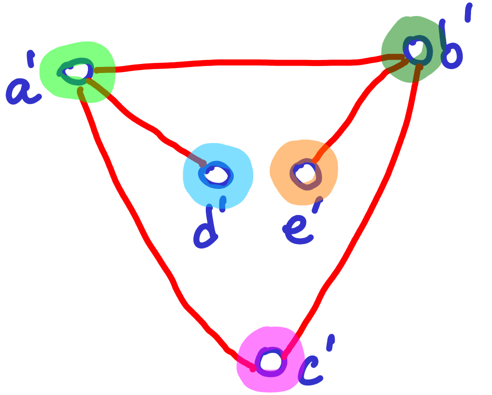
v	m(v)		
a	d'	w	r
b	a'	v	t
c	c'	x	u
d	b'	z	q
e	e'	y	s

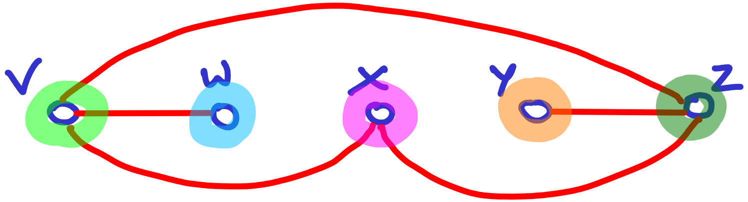
isomorphism → map vertices of graph G to vertices of graph H.

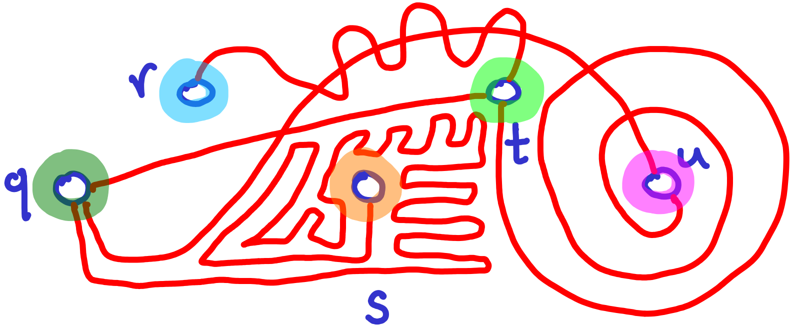
vertices $\alpha, \beta \in G$ share an edge in G

IFF vertices $m(\alpha), m(\beta) \in H$ share an edge in H.









a	d'	w	r
b	a'	v	t
c	c'	x	u
d	b'	z	q
e	e'	y	s

difficult – complicated – inefficient*

(not just because drawings look complicated)

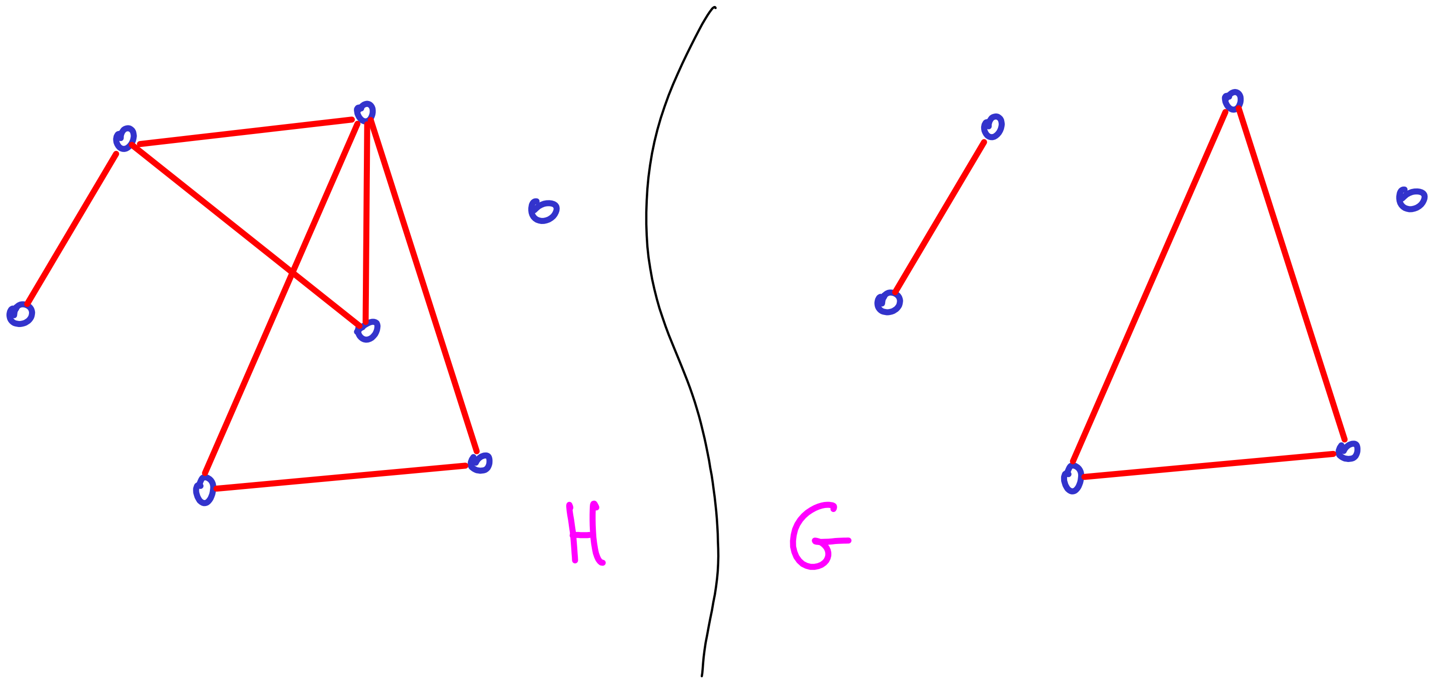
- Determining if two graphs are isomorphic (without a mapping)

- Counting # possible graphs without "doublecounting" isomorphs

* time complexity as function of V, E

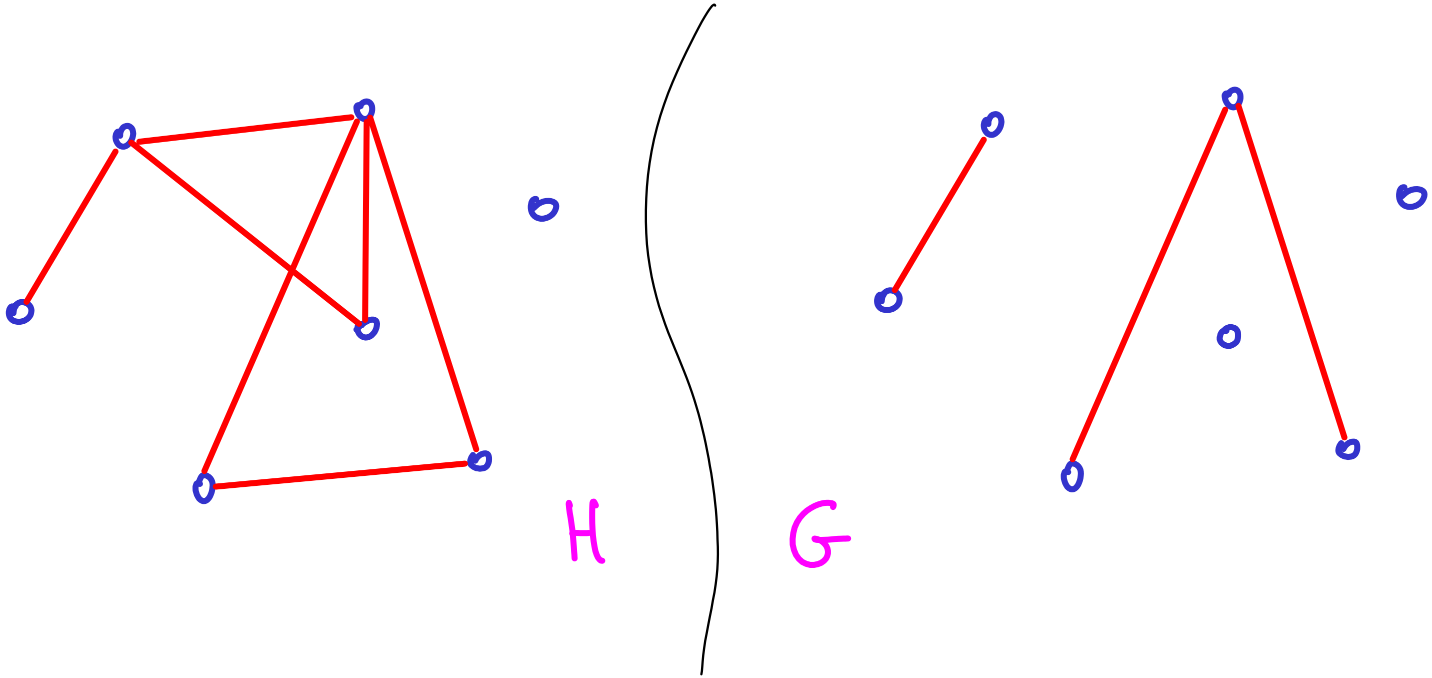
Subgraphs

G is a subgraph of H if $\begin{cases} V(G) \subseteq V(H) \\ E(G) \subseteq E(H) \end{cases}$



Subgraphs

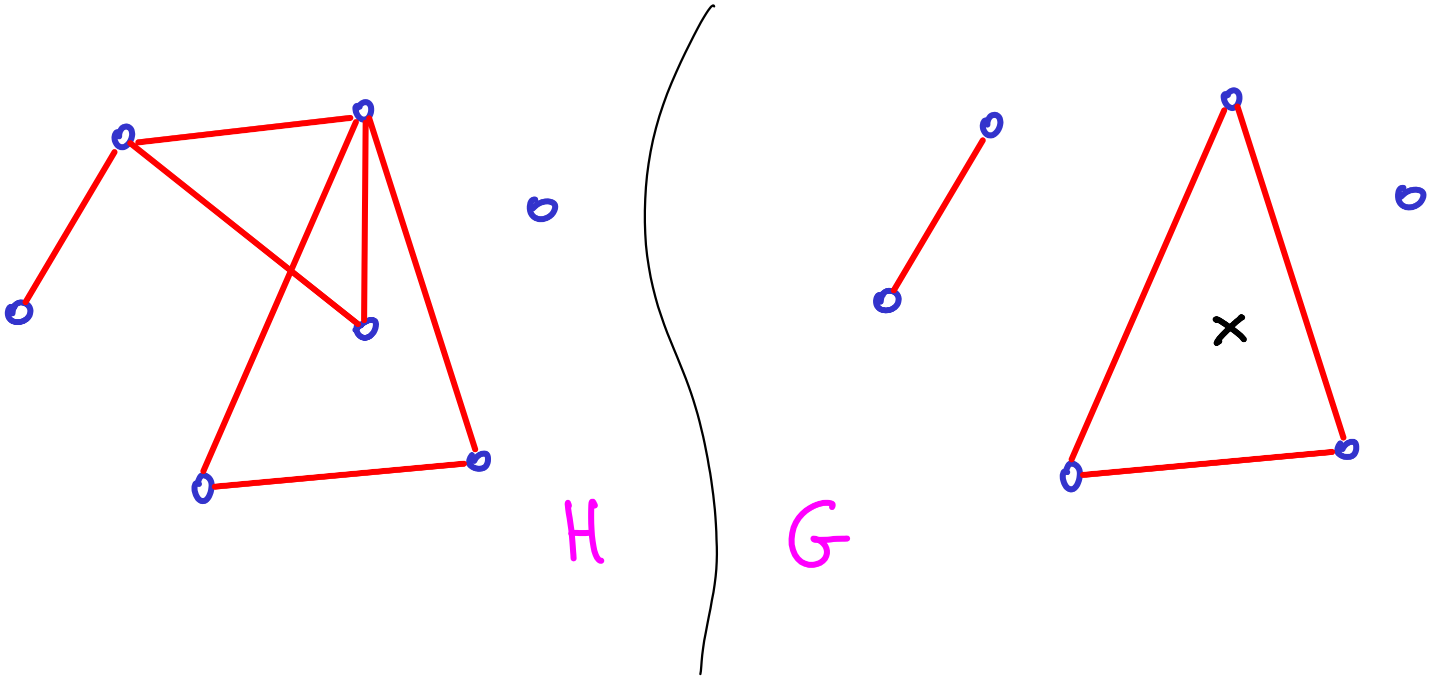
G is a subgraph of H if $\begin{cases} V(G) \subseteq V(H) \\ E(G) \subseteq E(H) \end{cases}$ if equal then it's a "spanning" subgraph



spanning subgraph of H

Subgraphs

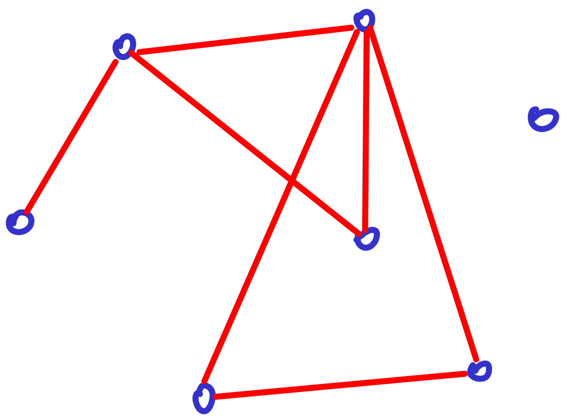
G is a subgraph of H if $\begin{cases} V(G) \subseteq V(H) \\ E(G) \subseteq E(H) \end{cases}$ if equal then it's a "spanning" subgraph



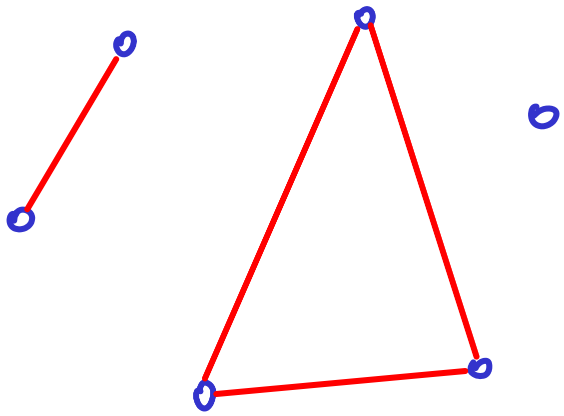
not a spanning subgraph of H

Subgraphs

G is a subgraph of H if $\begin{cases} V(G) \subseteq V(H) \\ E(G) \subseteq E(H) \end{cases}$ } if equal then it's a "spanning" subgraph



H



G

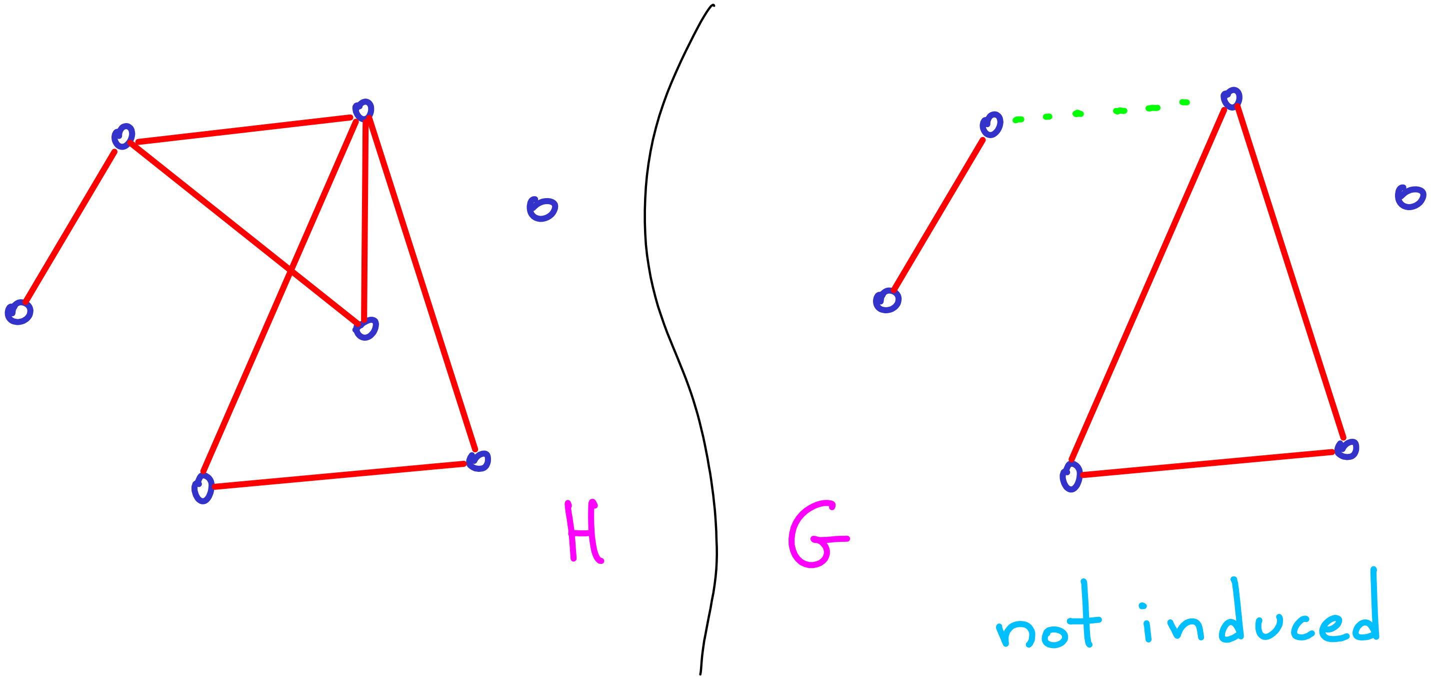
is this
↳ induced ?

If you only remove edges as a result of removing vertices then G is an "induced" subgraph

Subgraphs

$$G \text{ is a subgraph of } H \text{ if } \begin{cases} V(G) \subseteq V(H) \\ E(G) \subseteq E(H) \end{cases}$$

} if equal then it's a "spanning" subgraph



not induced

If you only remove edges as a result of removing vertices then G is an "induced" subgraph