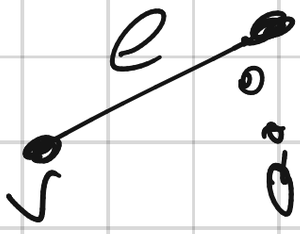
Graph $G = \langle V, E \rangle$

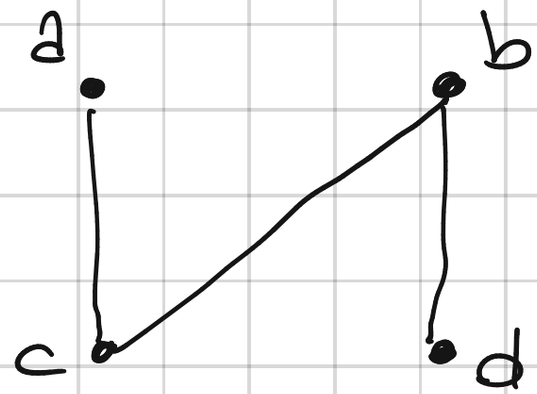
- V : vertices
- E : edges $\quad E \subseteq V \times V$

$(v, w) \in E \Rightarrow (w, v) \in E$

Undirected graph



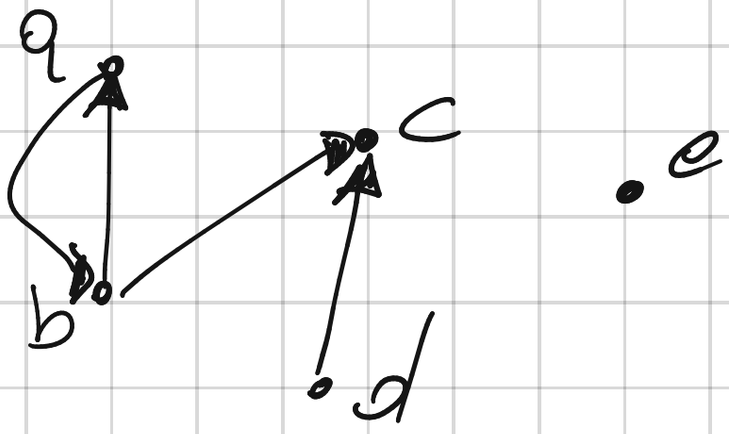
v incident on e



Graph $G = \langle V, A \rangle$

V : vertices

A : arcs $\quad A \subseteq V \times V$



e : isolated vertex

Graph $G = \langle V, E \rangle$

$|V| : n$ $\quad$ $|E| : m$

vertex encoded with $\Theta(\log n)$ bit

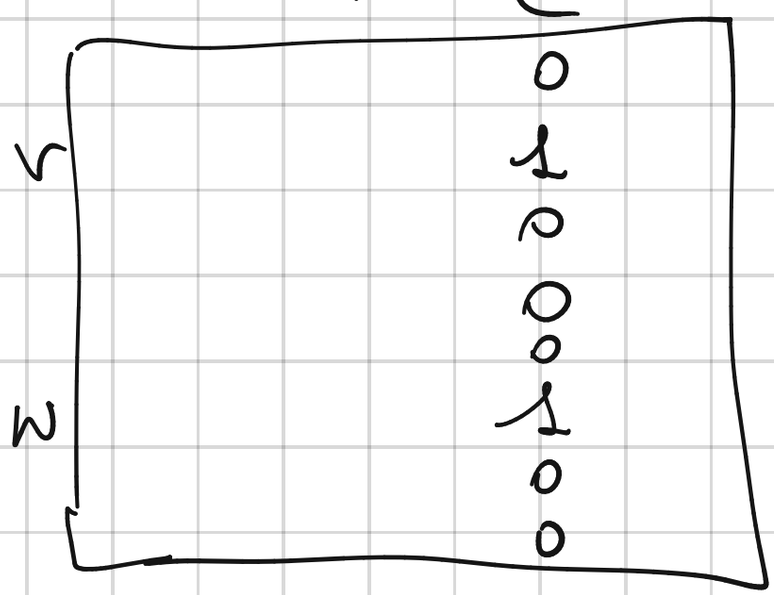
Access to a vertex in $\Theta(1)$ time

Graph Representations

- Adjacency Matrix $M[v,w]$ $O(n^2)$ space
  - given a vertex v, find edges incident on v
    $O(n)$ time
- Adjacency lists $L[v]$
  for each vertex v, $L[v]$ is the list of edges incident on v
  - given a pair of vertices v,w, is $(v,w) \in E$?
    $O(|N(v)|)$ time

- Incidence Matrix][V, e]

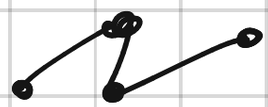
$\Theta(nm)$ space

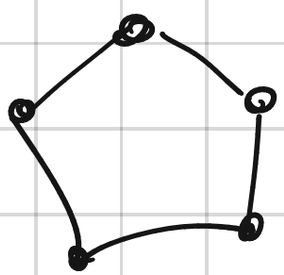


Special graphs

K_n : complete graph on n vertices

$K_{n,m}$: two sets A, B with $|A|=n$, $|B|=m$ and all possible edges in $A \times B$

P_n: simple path on n vertices P_4 

C_n: simple cycle on n vertices C_5 

Subgraph

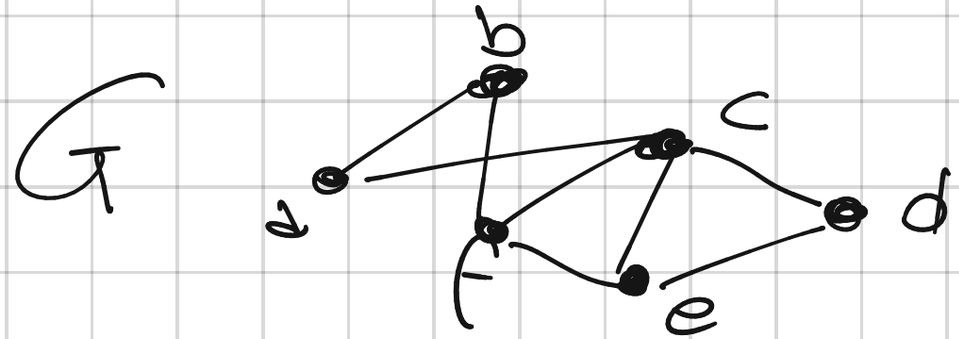
$G = \langle V, E \rangle$ $\quad$ $G_1 = \langle V_1, E_1 \rangle$

G_1 is subgraph of G if $V_1 \subseteq V$ and $E_1 \subseteq E$

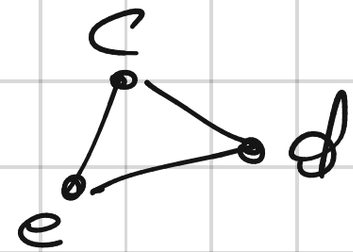
Induced subgraph $G|W$ with $U \subseteq V$

$G = \langle V, E \rangle$

$G|W = \langle W, E \cap (W \times W) \rangle$



$G|\{c,d,e\}$



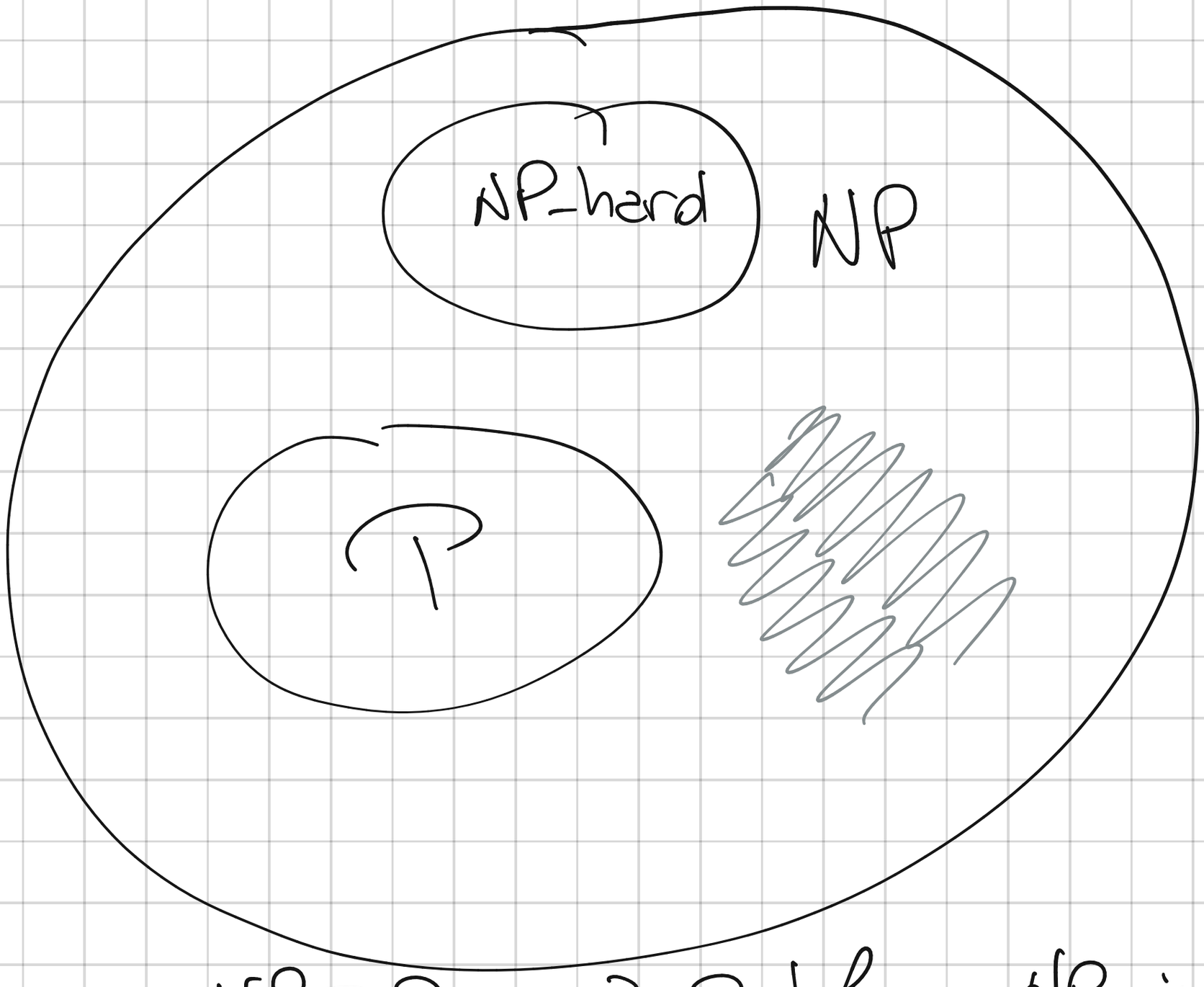
Isomorphism

$G_1 = \langle V_1, E_1 \rangle$ $\quad$ $G_2 = \langle V_2, E_2 \rangle$

$\varphi : V_1 \mapsto V_2$ such that $(\varphi(v), \varphi(w)) \in E_2 \iff (v, w) \in E_1$

Automorphism

if $G_1 = G_2$



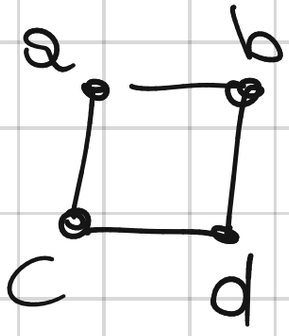
$NP \supset P \Rightarrow \exists$ Problem NP_ intermediate

Product

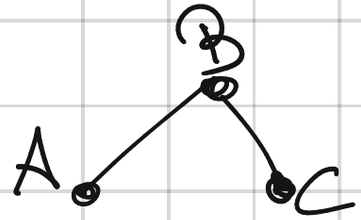
$G_1 = \langle V_1, E_1 \rangle$ $\quad$ $G_2 = \langle V_2, E_2 \rangle$

$$G_1 \times G_2 = \langle V_1 \cup V_2, E_1 \cup E_2 \cup (V_1 \times V_2) \rangle$$

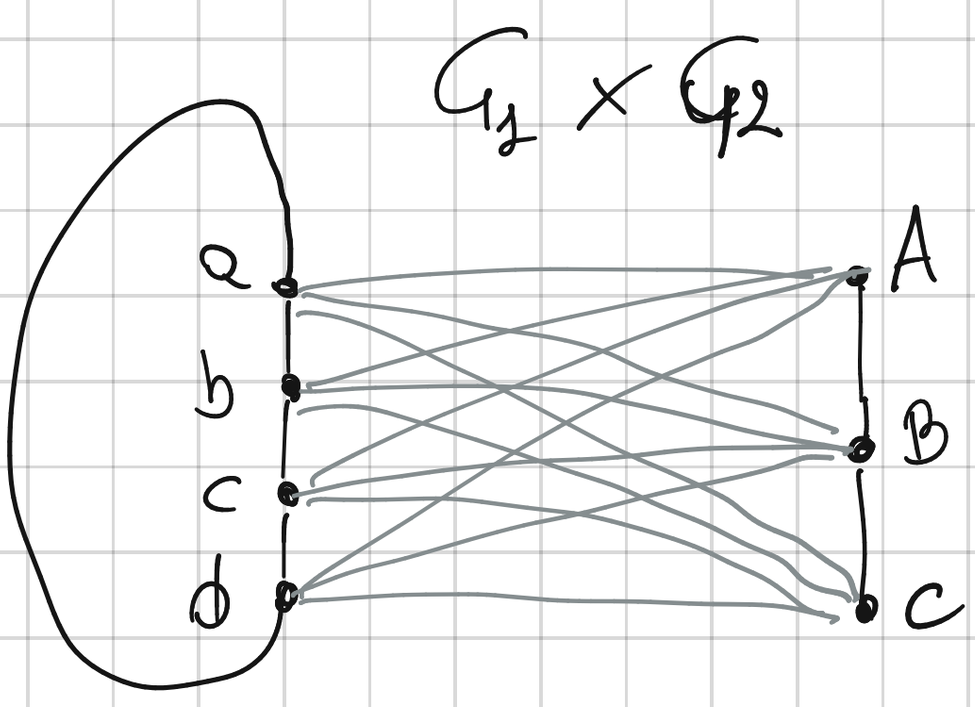
G_1



$G_2 =$



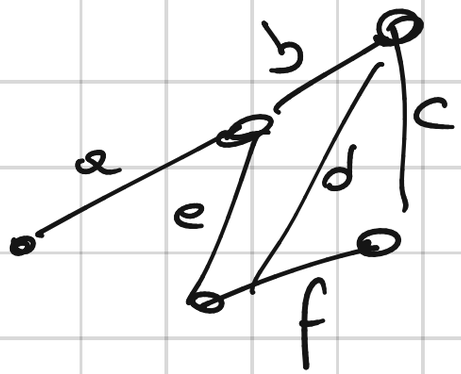
$G_1 \times G_2$

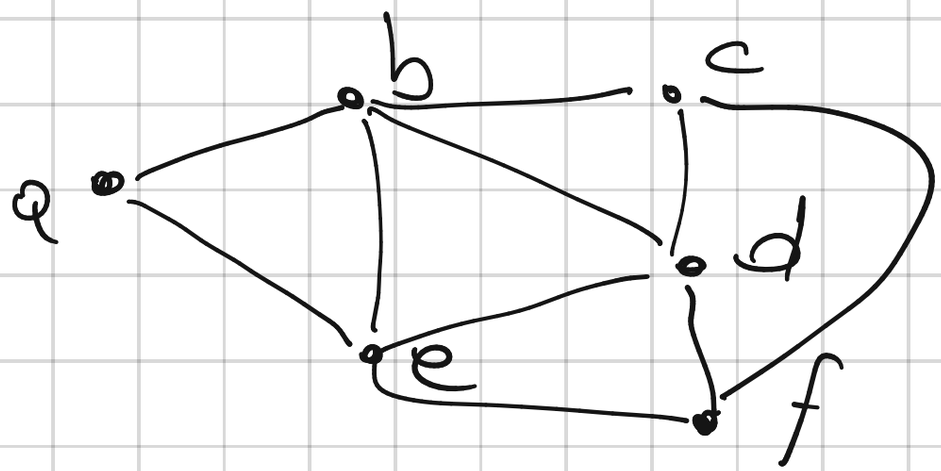


Complement $G=\langle V, E\rangle$ $\quad \overline{G}=\langle V, V\times V \setminus E\rangle$

Line graph $L(G)$ $\quad G=\langle V, E\rangle$

$L(G) = \langle E, \}(e_1, e_2)$ such that e_1 and e_2 share an endpoint in $G\}\rangle$





Neighborhood

$$N(v) = \{w : (v,w) \in E\}$$

$$N^+(v) = \{w : (v,w) \in A\}$$

$$N^-(v) = \{w : (w,v) \in A\}$$

] Directed graphs

$|N(v)| =$ degree

k-Regular if $N(v) = k \quad \forall v \in V$

Exercise

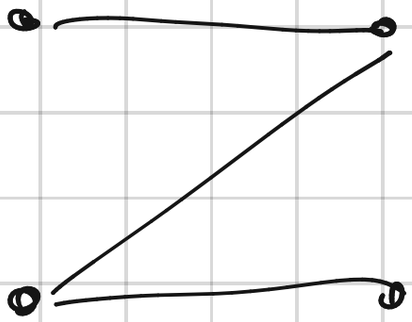
Find a graph G that is the complement of itself.

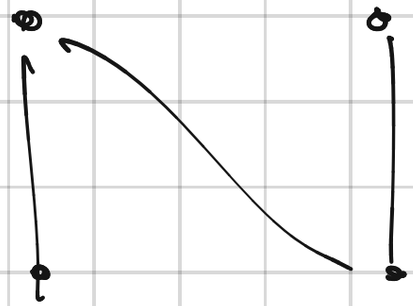
Ex 2

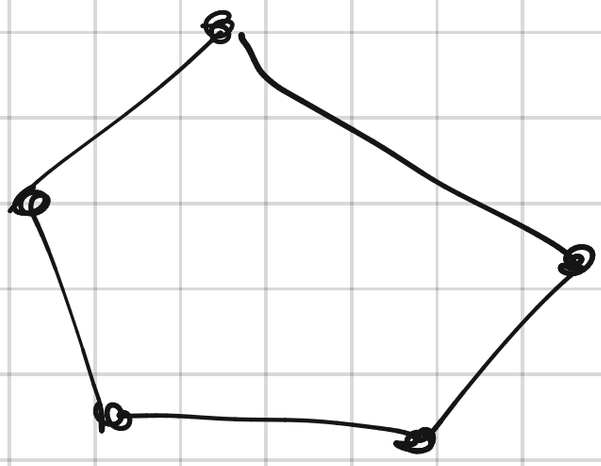
Find a 3-regular (cubic) graph that is the complement of itself

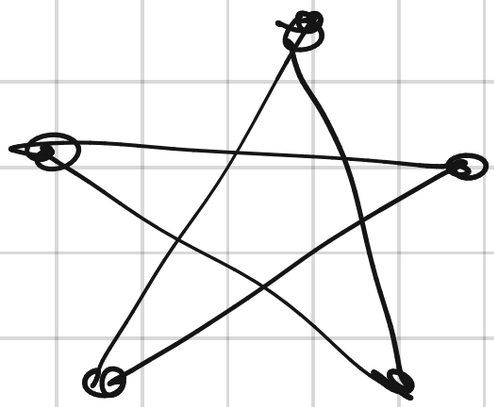
Exercise

$G = \overline{G}$









Exercise 2

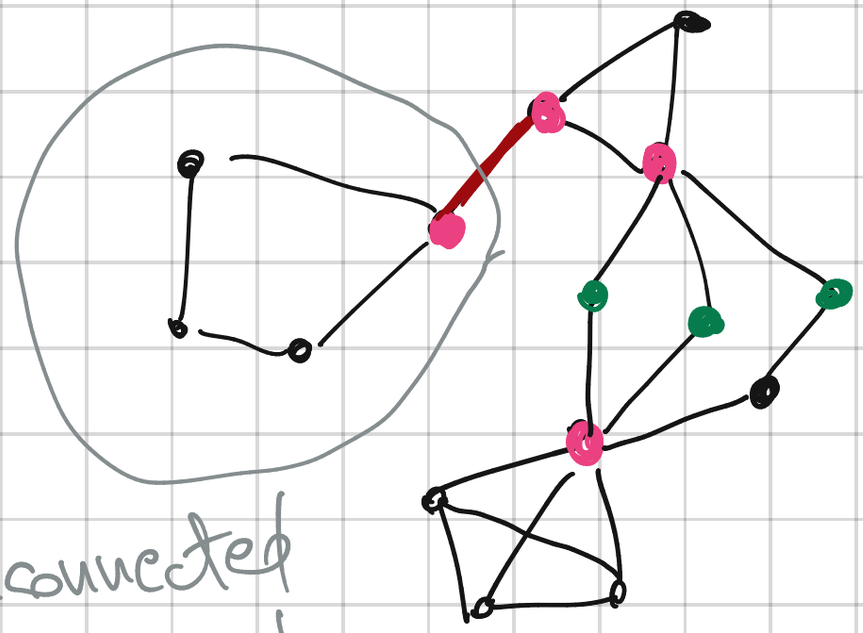
1) Cubic graph has an even number of vertices (n)

2) $v \in V$ has 3 neighbors and $n-4$ non-neighbors

(v itself cannot be a neighbor)

$n-4$ is even $\neq 3$

Connected graph



biconnected component

- articulation point / cutvertex
- bridge
- cutset (of size 3)

Connectivity ≤ min degree

Connectivity = min size of a cutset

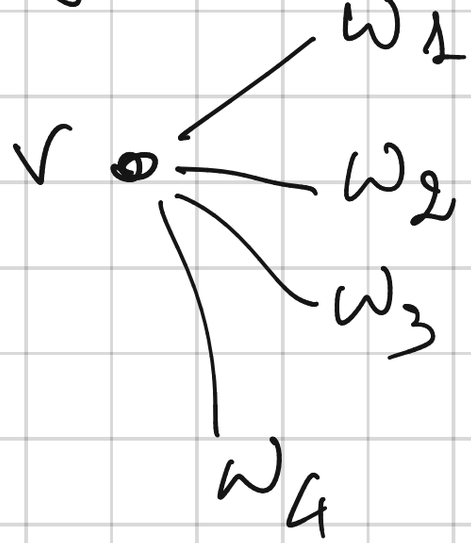
Ex

1) Prove that connectivity $\leq$ min degree

2) Find a graph with connectivity $<<$ min degree

Ex 1

Let v be a vetex with smallest neighborhood.

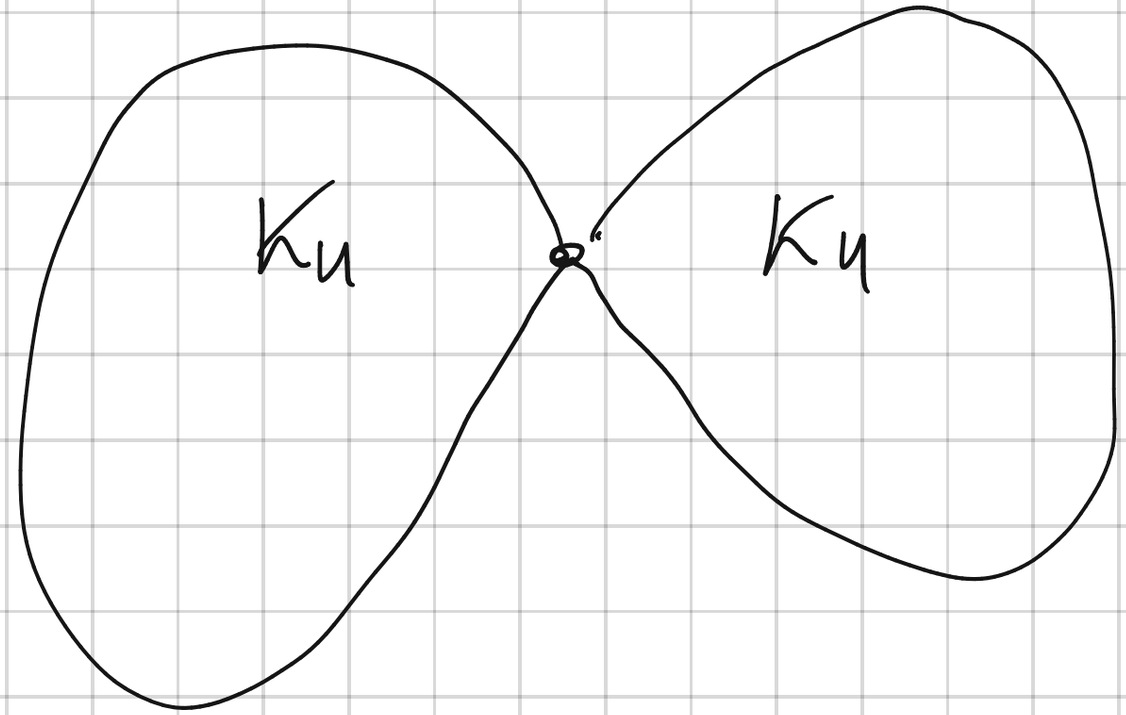


Remove all w_i

$\Downarrow$

v is isolated

Ex2



Two K_4 sharing a vertex

Tree

- acyclic and connected
- minimally connected
- maximally acyclic

Depth-first Breadth-first visits

- Compute connected components
- Compute biconnected components
- Topological sort
- Shortest path

Depth-first visit (DFS)

L empty list (unvisited vertices)

Pick any unvisited vertex v

Visit(v)

Visit(v)

Add v to the end of L

For each unvisited w such that $(v,w) \in E$

Visit(w)

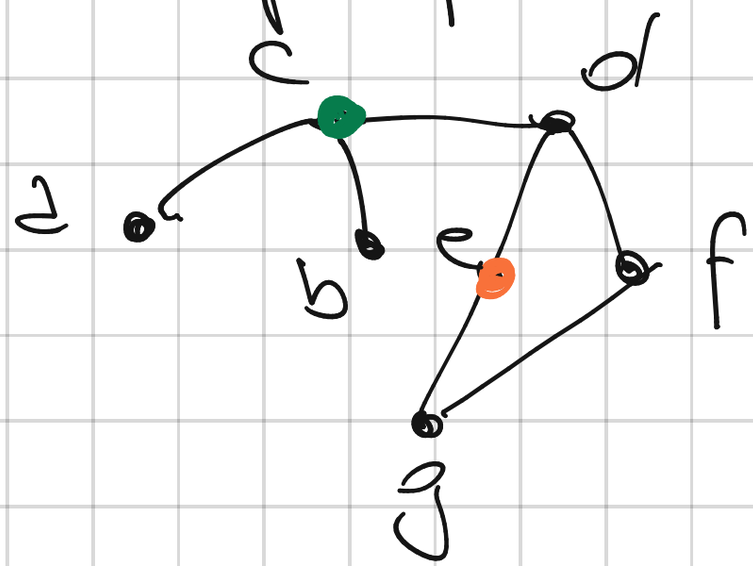
Breadth-first visit (BFS)

```
Visit(v)
Add v to the end of L (if v is not in L)
W := set of unvisited vertex w s.t. (v,w)∈E
For each w∈W
    Add w to the end of L
For each w∈W
    Visit(w)
```

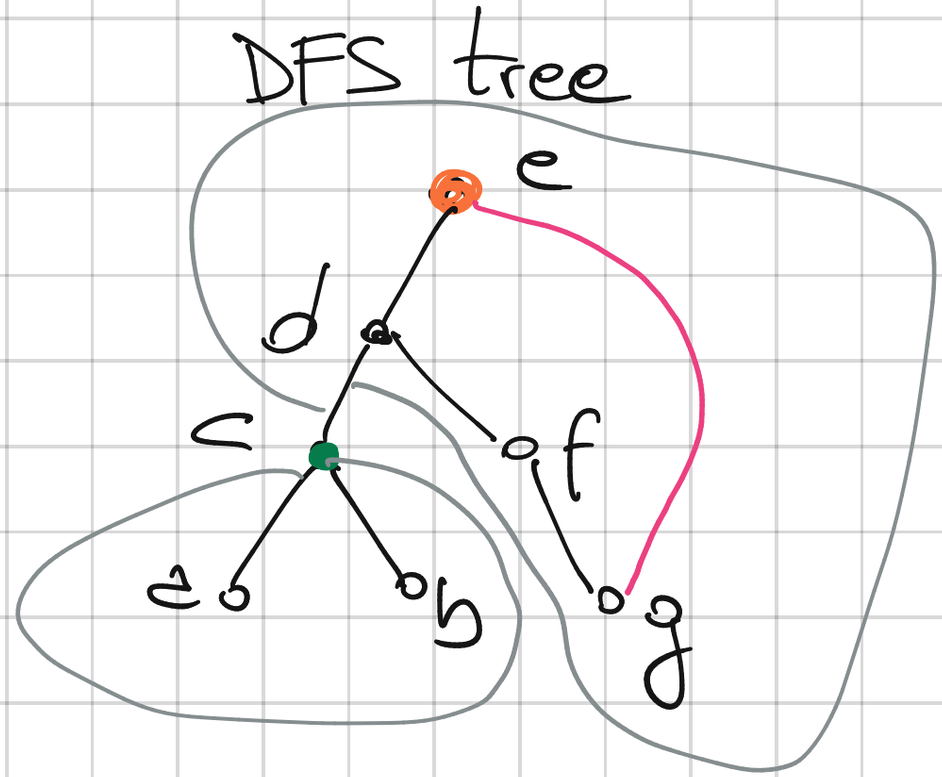
Question

Find all articulation vertices of a graph G?

DFS depth-first visit



articulation vertex

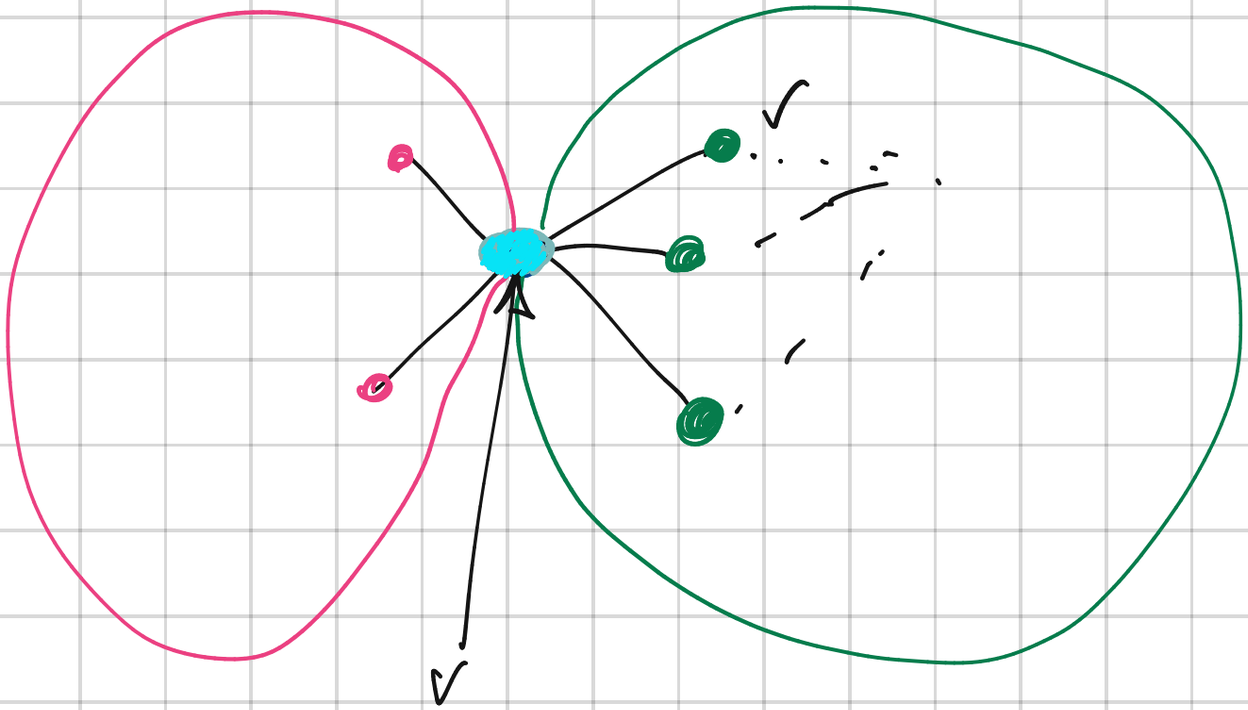


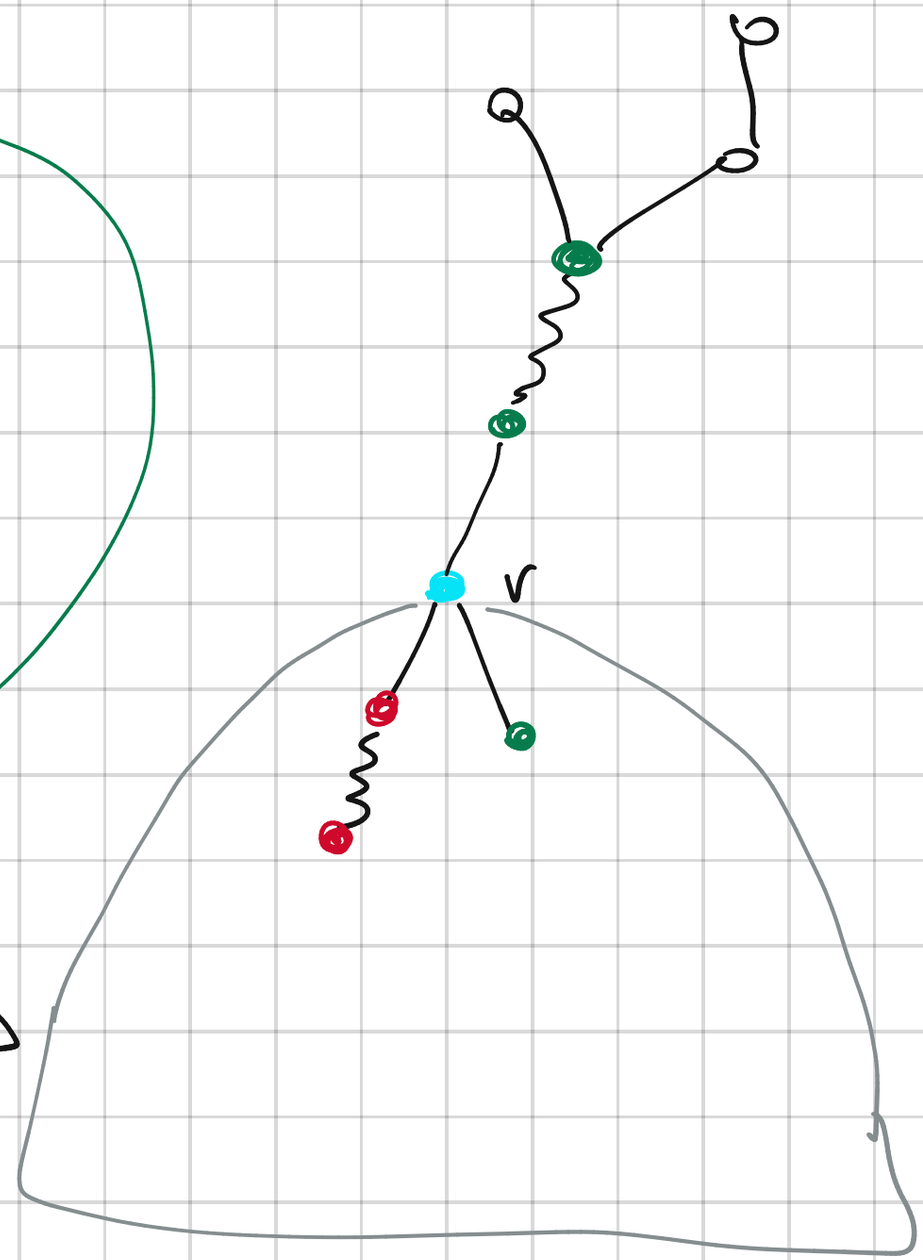
Compute biconnected components

- lowpoint(v) : min depth of all neighbors of descendents of v

v is an articulation point iff lowpoint(c) ≥ depth(v) with c a child of v in DFS

root of DFS is a special case





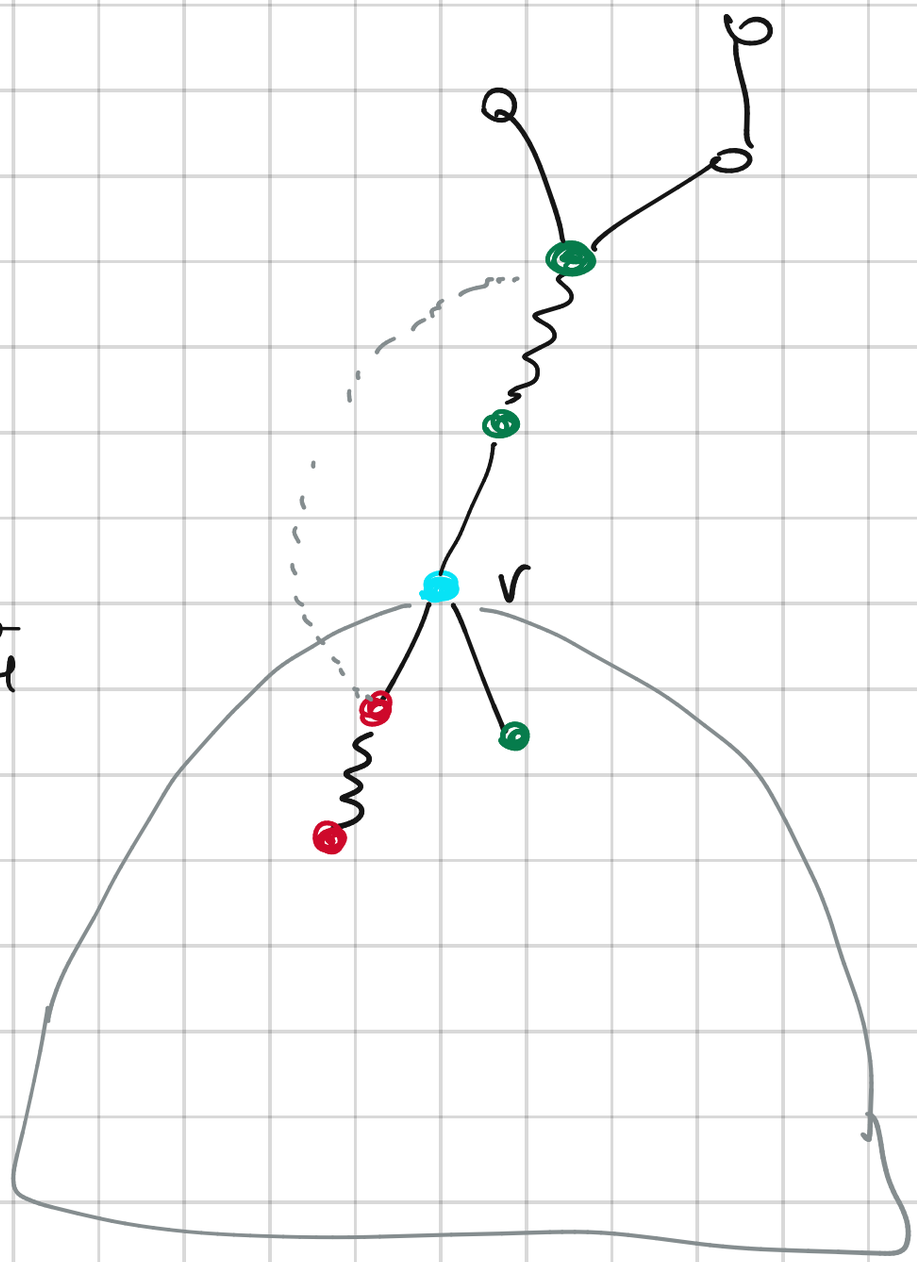
1) All red vertices are →

2) Red and green vertices are separated

Assume now v is not an articulation point

Then there is a second path from $\bullet$ to $\bullet$ without going through v.

Consider the last vertex of such path that is below v $\Rightarrow$ it is adjacent to a vertex above v $\Rightarrow$ depth of lowpoint of $\bullet$ is smaller than that of $\bullet$



Topological Sort

L := empty list (contains the sorted vertices)

While there exists at least an unvisited source s

    Add s to the end of L

    Remove all arcs outgoing from s

Topological sort via DFS

L := empty list

While there is an unmarked vertex v

visit(v)

Visit(v)

if v is visited => Return

For each w s.t. (v,w) is an arc

Visit(w)

Add v to the head of L

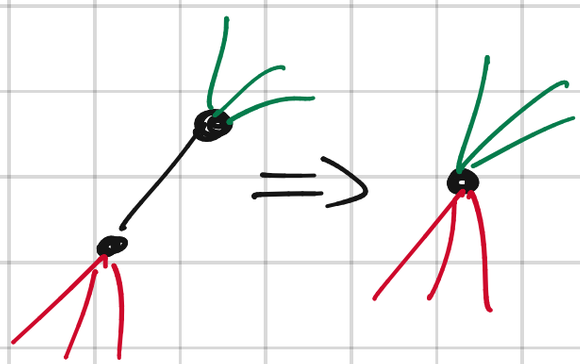
Graph classes

- Nested graphs
- Series-parallel graphs

Graph minor

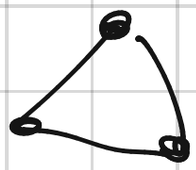
H is a minor of G if H can be obtained from G via:

1. deleting vertices
2. deleting edges
3. contracting edges



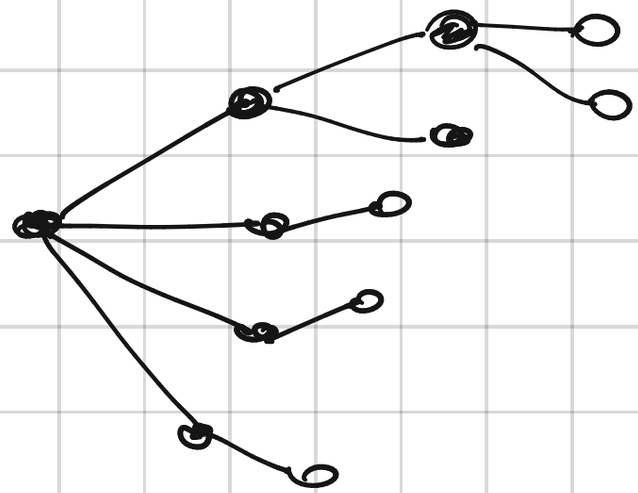
Robertson - Seymour Thm

Let P be a property that is preserved by minor.
Then there exist a finite set F of forbidden minors such that a graph G has P iff no minor of G is in F.

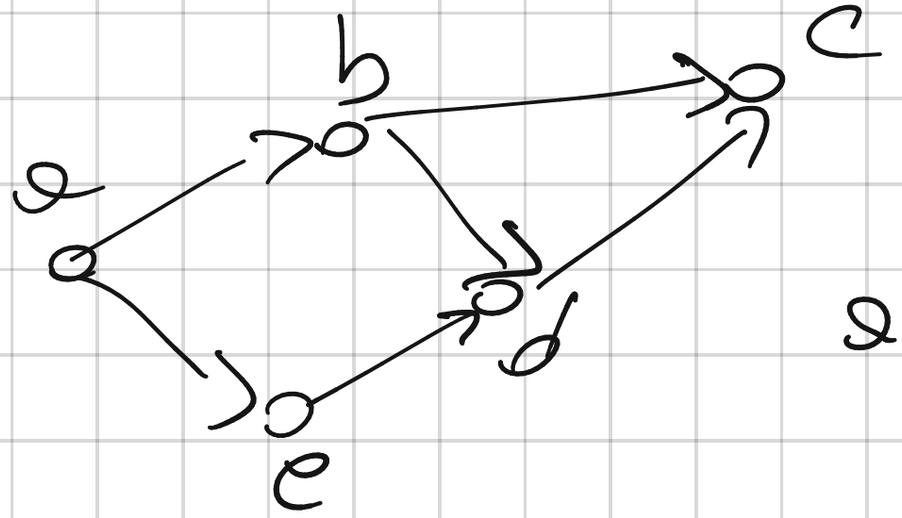
- Forests 

Given F => Polytime algorithm

BFS breadth-first visit



- compute the connected components
- shortest path from v to w
- Topological sort (given a dag G - dag = directed acyclic graph - find an order of V consistent with the arcs of G)



abedc ← Topological Sort

$N^-(a) = \emptyset$ source

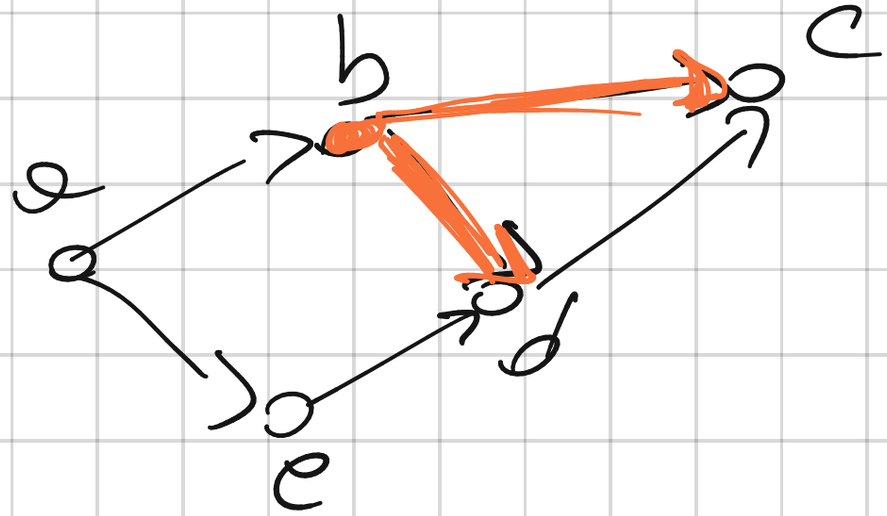
$N^+(r)$ sink

L := empty list

While there is a source s

Put s at the end of L

Remove all arcs outgoing from s



A = DFS(r) A is the set of vertices reached from r with the DFS(r) (excluded r)

b | (c, d)

DFS(d)

DFS(a) a | (c, d)

$L :=$ empty list
While there is a vertex v that is not in L:
    Visit(v)

Visit(v)

if v is in L => Return

For each w such that (v,w) is an arc
    Visit(w)
Add v to the head of L

BFS (v)

queue Q := (v)

All vertices in V are unmarked, except for v

While Q is not empty

Extract the first element x of Q

For each unmarked vertex w s.t. $(v, w) \in E$

Mark w

Add w to the end of Q

BFS(v)

- Maintaing a queue Q
- extract from Q a vertex x s.t.
  there exists an edge (w,x) with w
  output the earliest.

Lex-BFS(v)

- extract from Q the vertex x whose set of predecessors is lexicographically smallest

$S = ((v_1, v_2, \ldots, v_n))$

1) Pick the first vertex v from the first set

Split each set X in S into $X \cap N(v)$, $X \setminus N(v)$

$S = ((v_3, v_5 \ldots)(v_2, v_4, \ldots))$

$N(v)$ $\quad$ $\overline{N(v)}$

BFS maintain a list T of visited vertices

Let $r, w \in V$ at a certain point in time

T

$r \rightarrow N(r) \cap T$

$w \rightarrow N(r) \cap T$

$T = v(t_1)\ v(t_2)\ v(t_3) \ldots\ v(t_k)$

$r = 1 \quad 0 \quad 1 \quad 1$

$w = 1 \quad 1 \quad 0 \quad 0$

$v(t_1) \in N(w)$

$v(t_3) \notin N(w)$

Graph classes

- connected graphs
- acyclic graph
- trees

Recognizing a graph class $\mathcal{C}$

Instance: a graph G

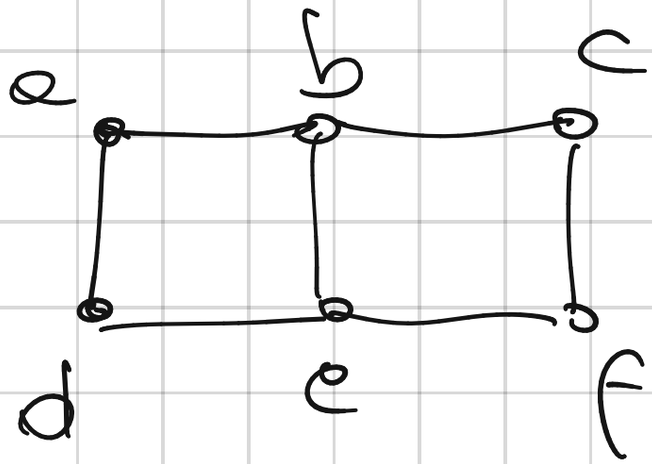
Question: does G belong to the class $\mathcal{C}$?

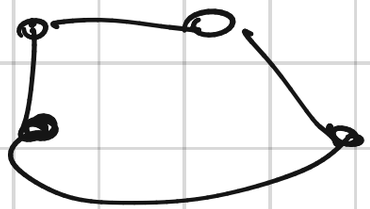
Examples of graph classes

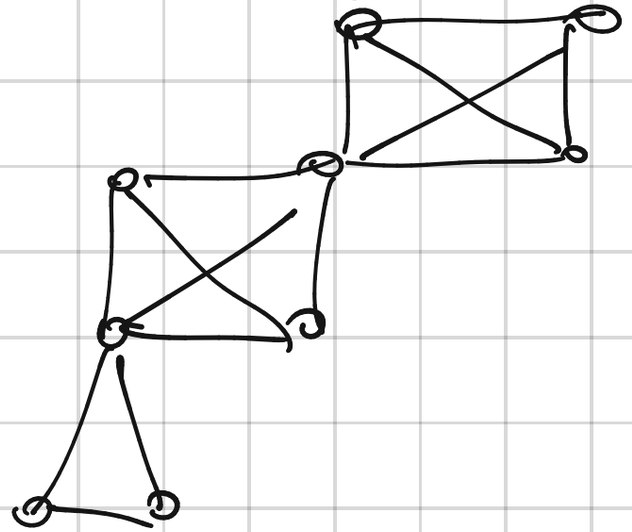
- Interval graphs

  Vertices → intervals on the line of reals

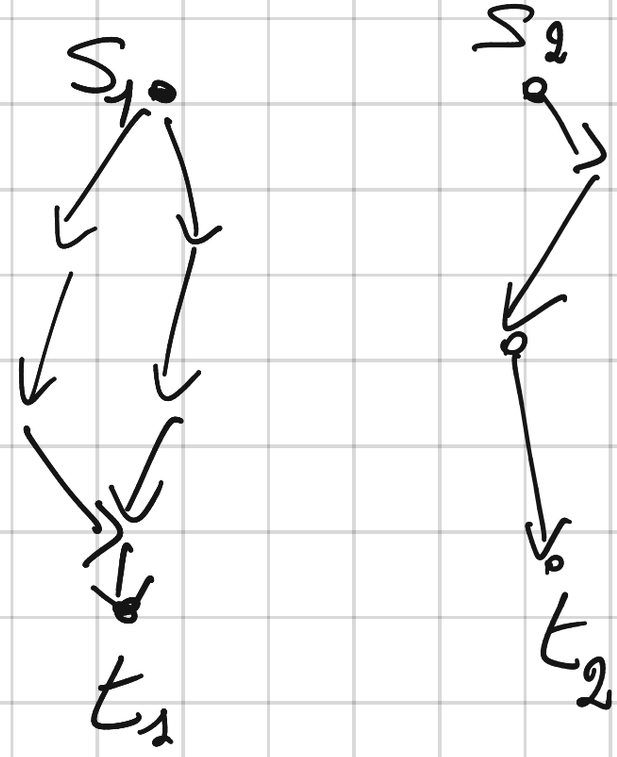
  Edges → overlapping intervals



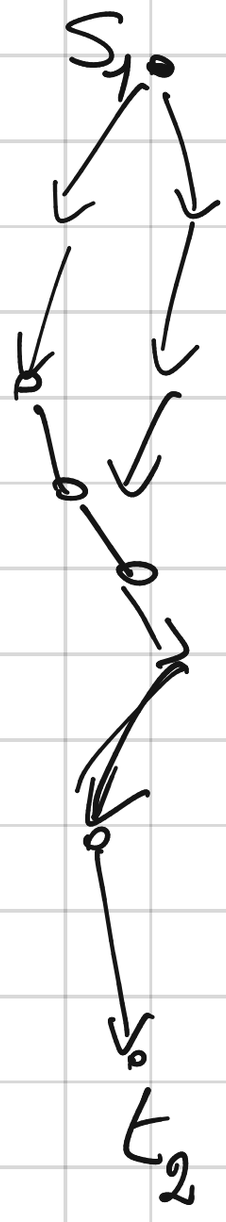




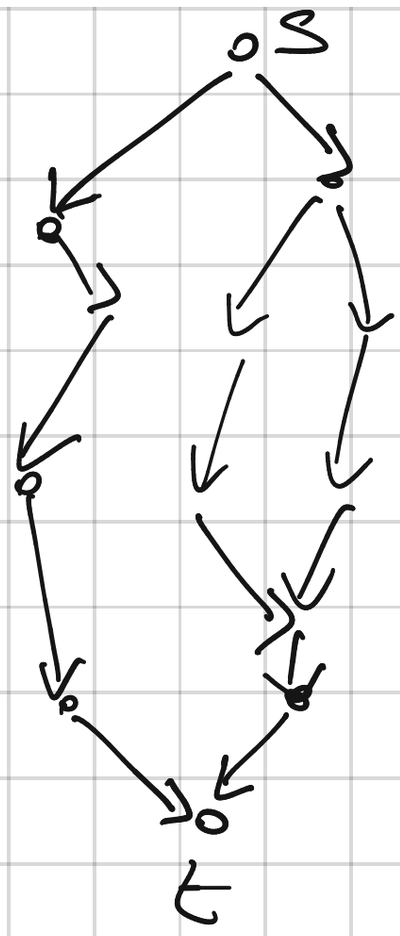
- seres-paralel graphs



$s \circ \rightarrow \circ t$

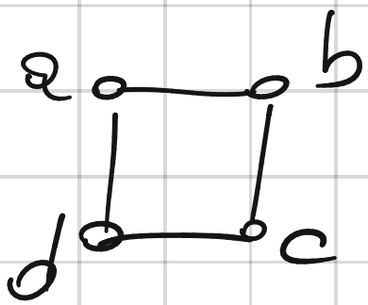


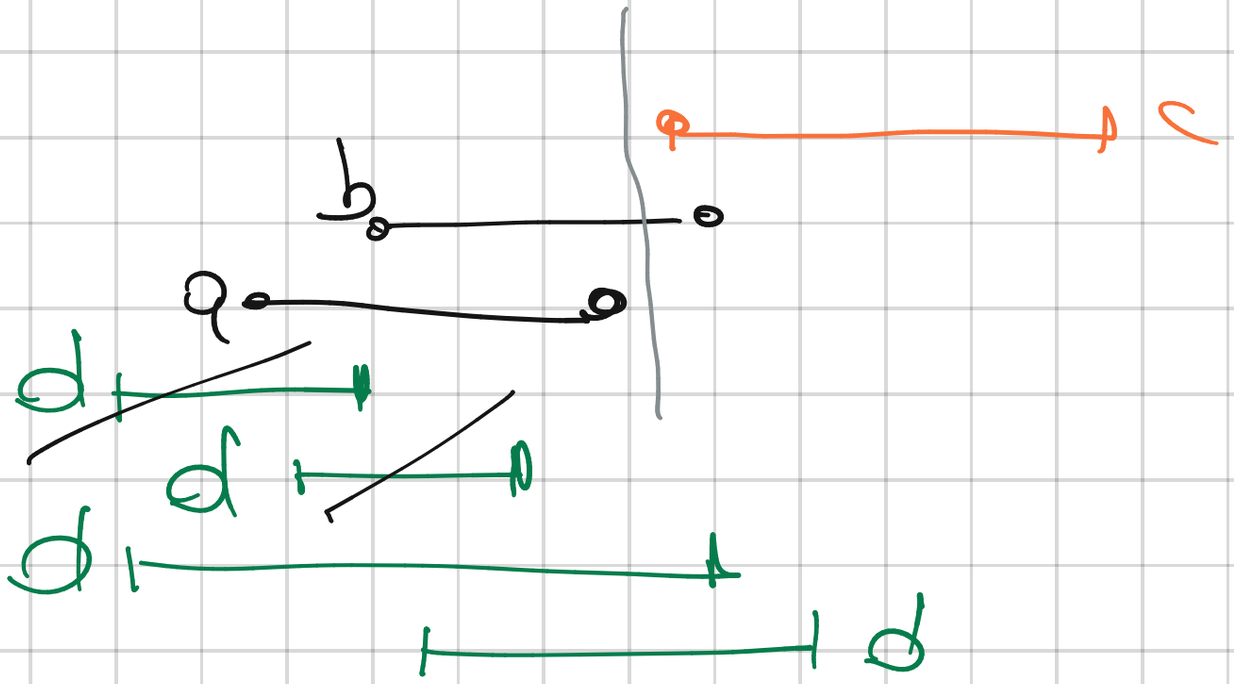
Serie



paralleb

Proof that C_4 is not an interval graph





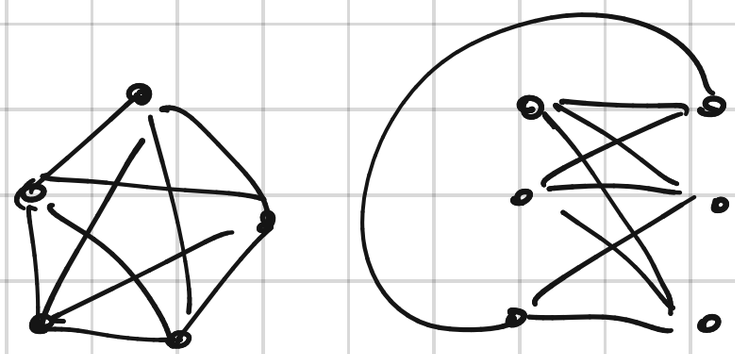
Planar graphs

G is a planar graph if it is possible to embed the vertices of G in the plane such that no two edges cross.

Wagner - Kuratowski thm

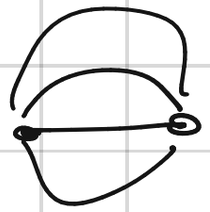
G is planar iff it does not have

K_5 or $K_{3,3}$ as minor



SPQR tree

S: series => a cycle with a virtual edge



P: parallel => a dipol

Q: a single edge

R: rigid subgraph