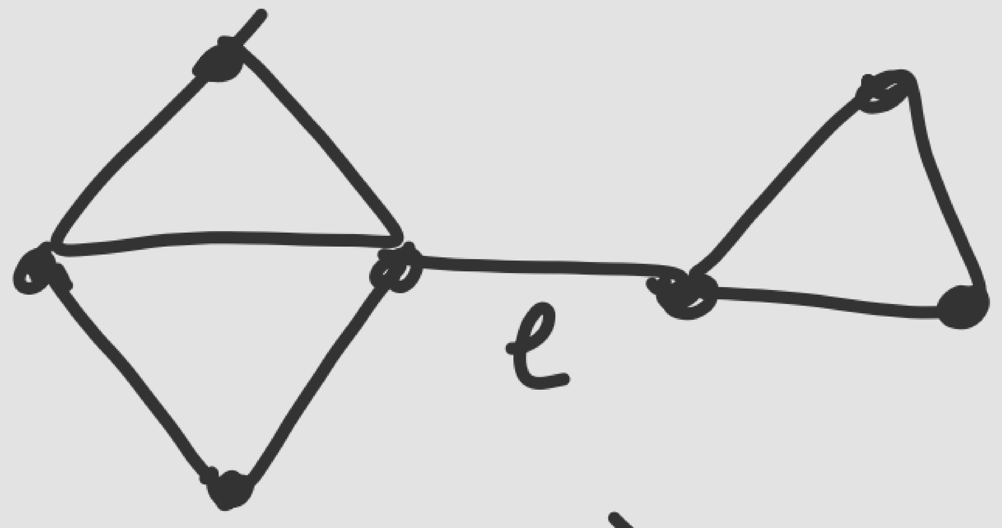
Connected Graph

loopless



Edge connectivity

Vertex Connectivity

Edge connectivity $\lambda(G)$

minimum number of edges we need to remove to make it disconnected / trivial

$$c(G) < c(G-e)$$

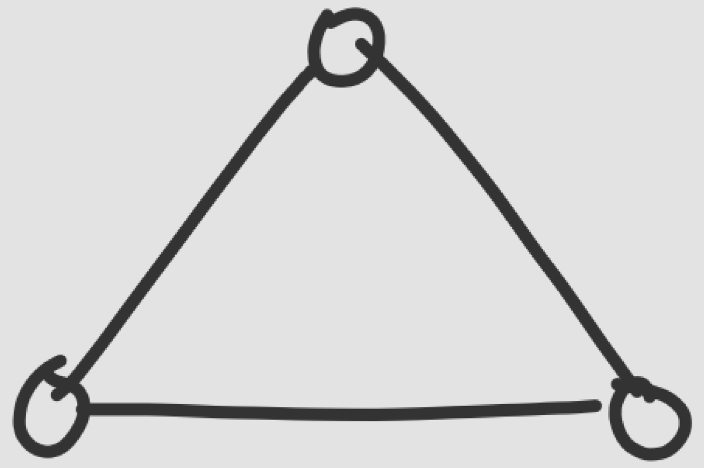
Degree (min) of a graph

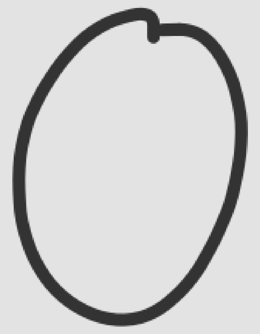
$\delta(G)$

Vertex with minimum degree

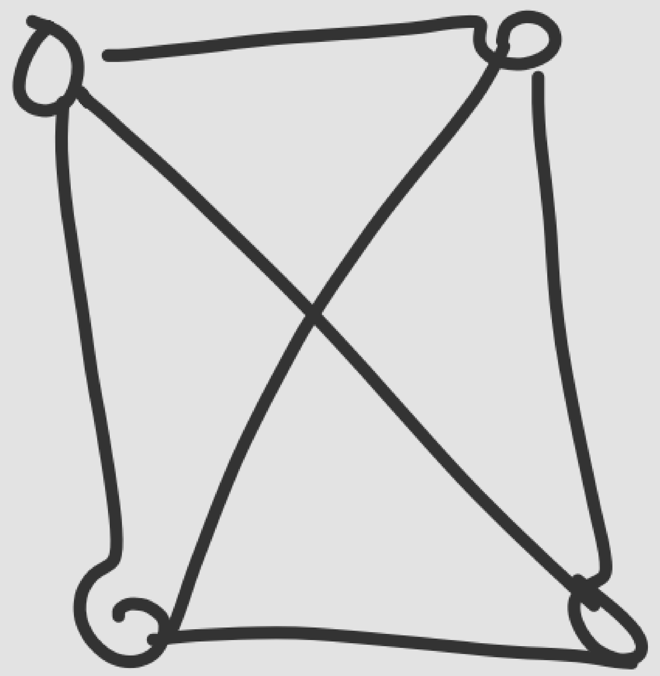
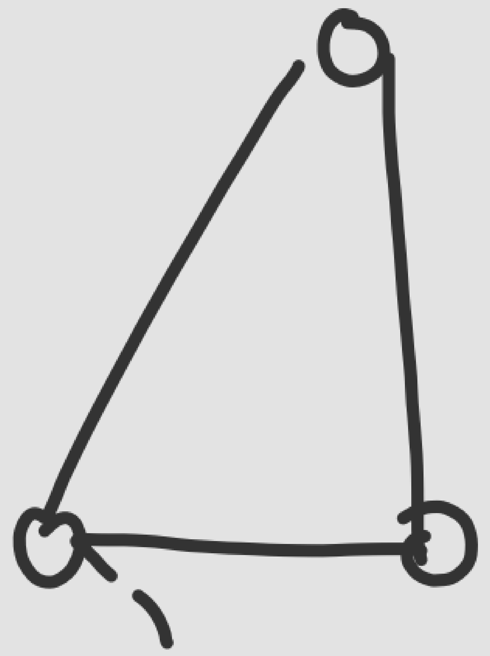
Vertex Connectivity, $\kappa(G)$ {kappa}

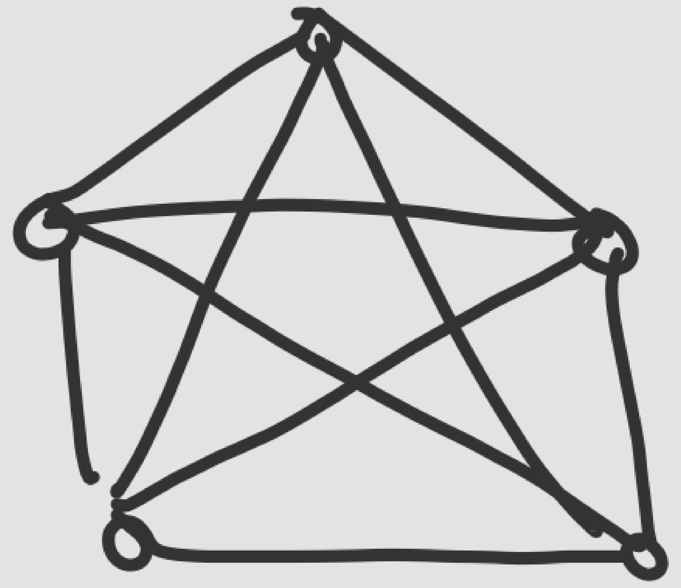
minimum number of vertices that we need to remove to make a graph disconnected or make it **trivial**

K_3:  $\kappa(K_3) = 2$

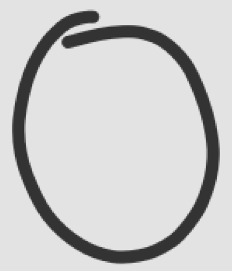


G:

K_4:  $\kappa(K_4) = 3$ 

K_5: 

$\kappa(K_n) = n-1$



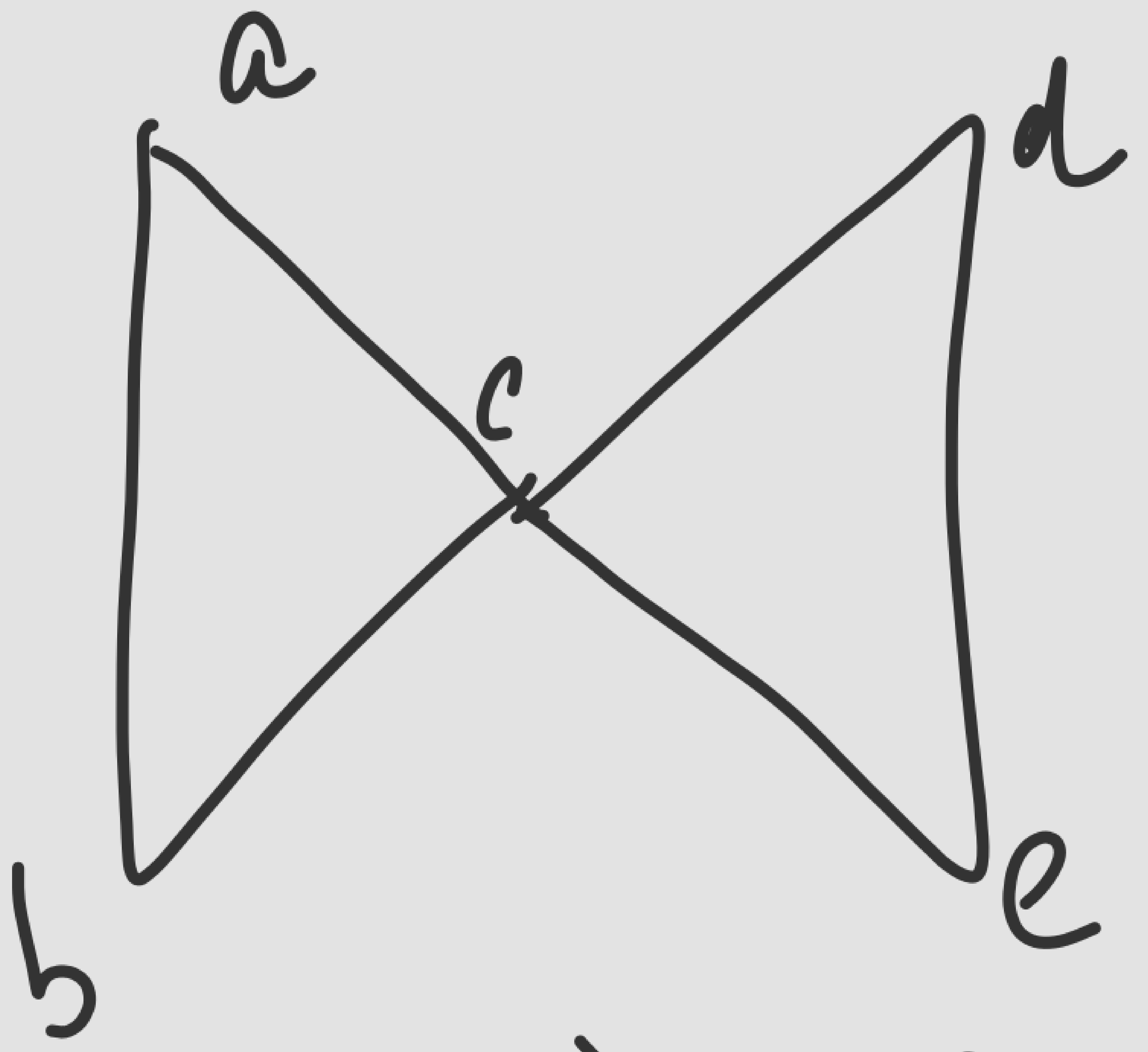
u

$\kappa(G) = 0$

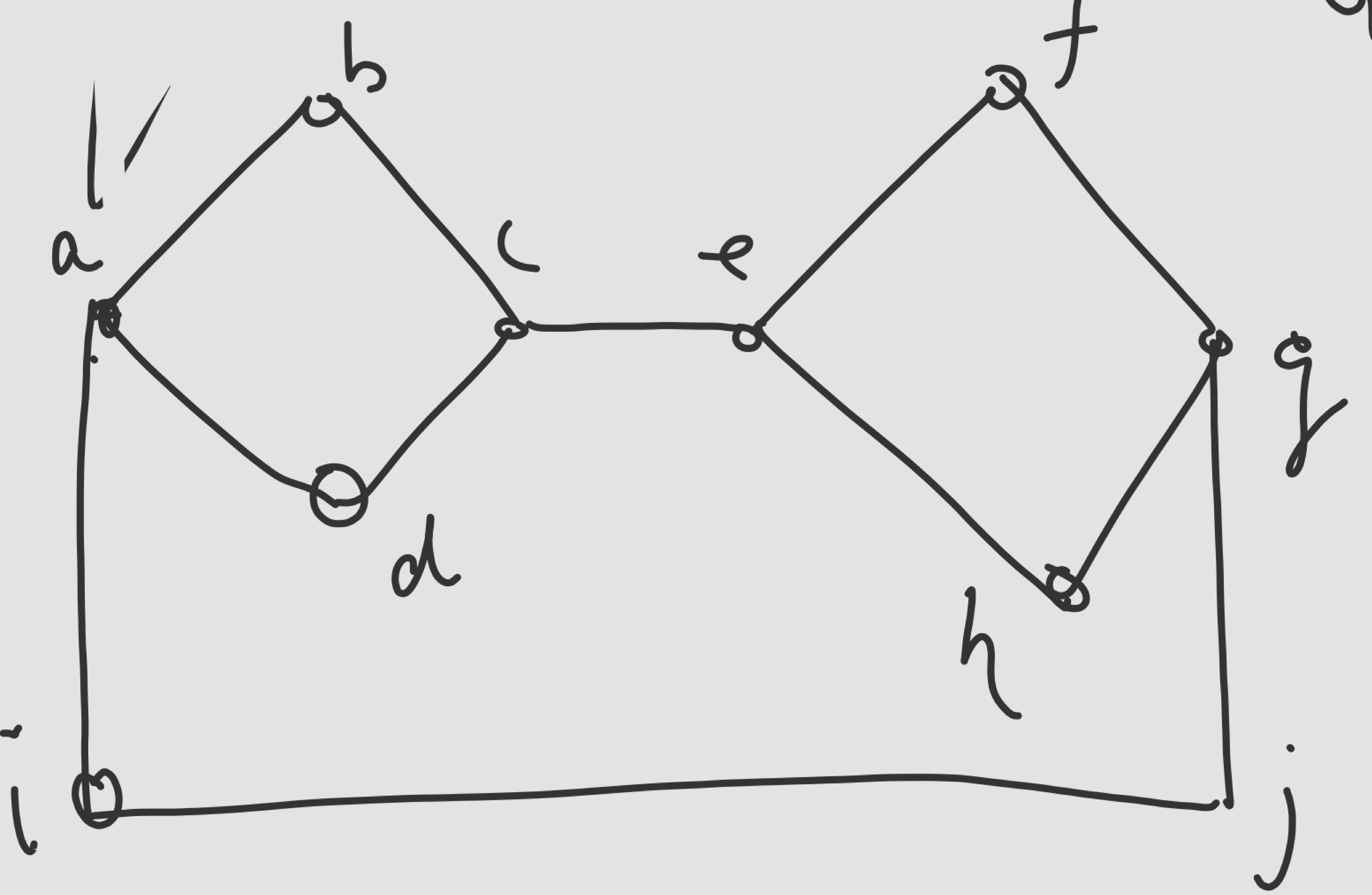
$$\boxed{0 \leq \kappa(G) \leq n-1}$$

In General,

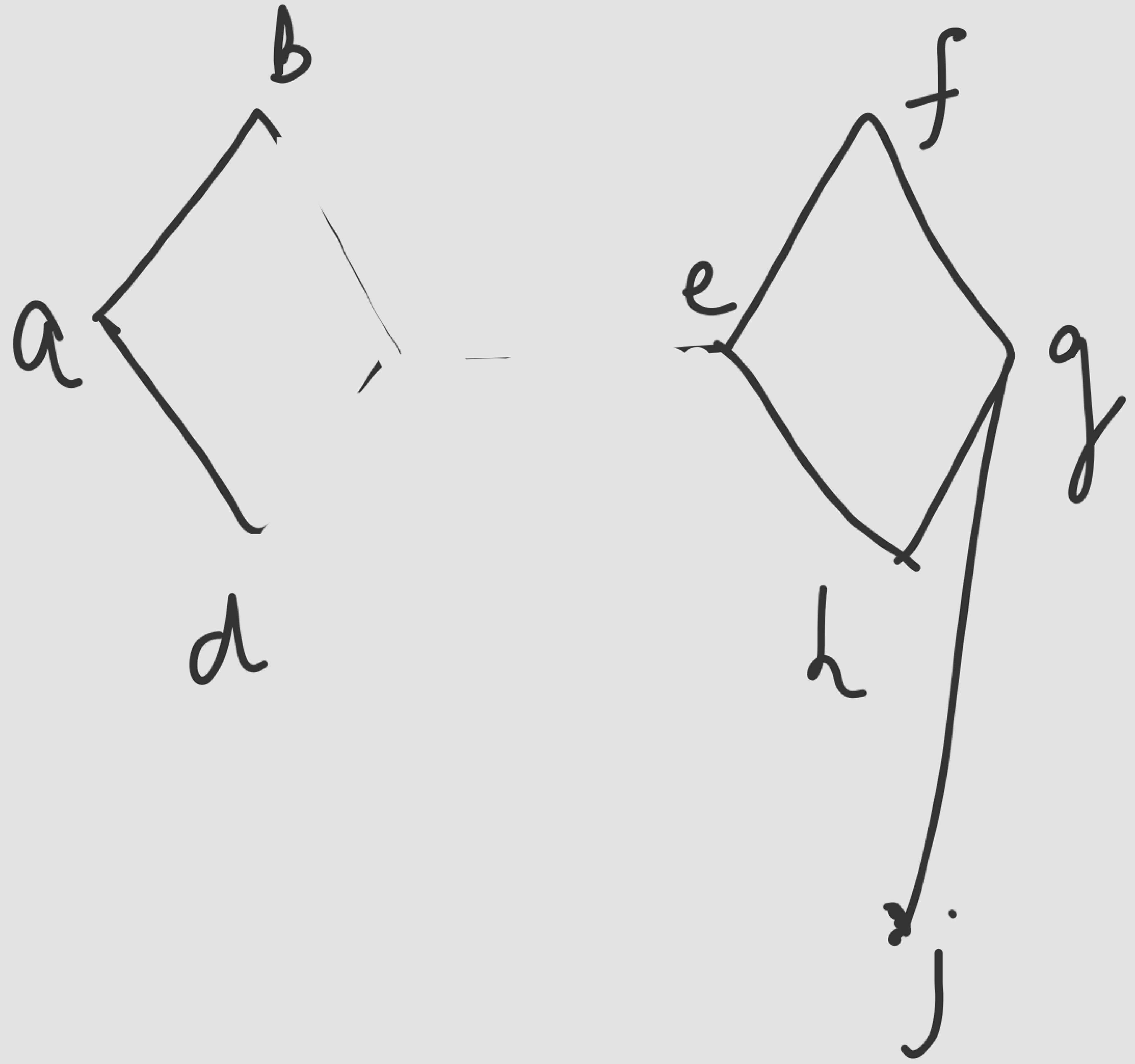
$$\underbrace{\kappa(G)}_{\text{V. Connectivity}} \leq \underbrace{\lambda(G)}_{\text{Edge connectivity}} \leq \underbrace{\delta(G)}_{\text{min degree}}$$



$\lambda(G) = 2$

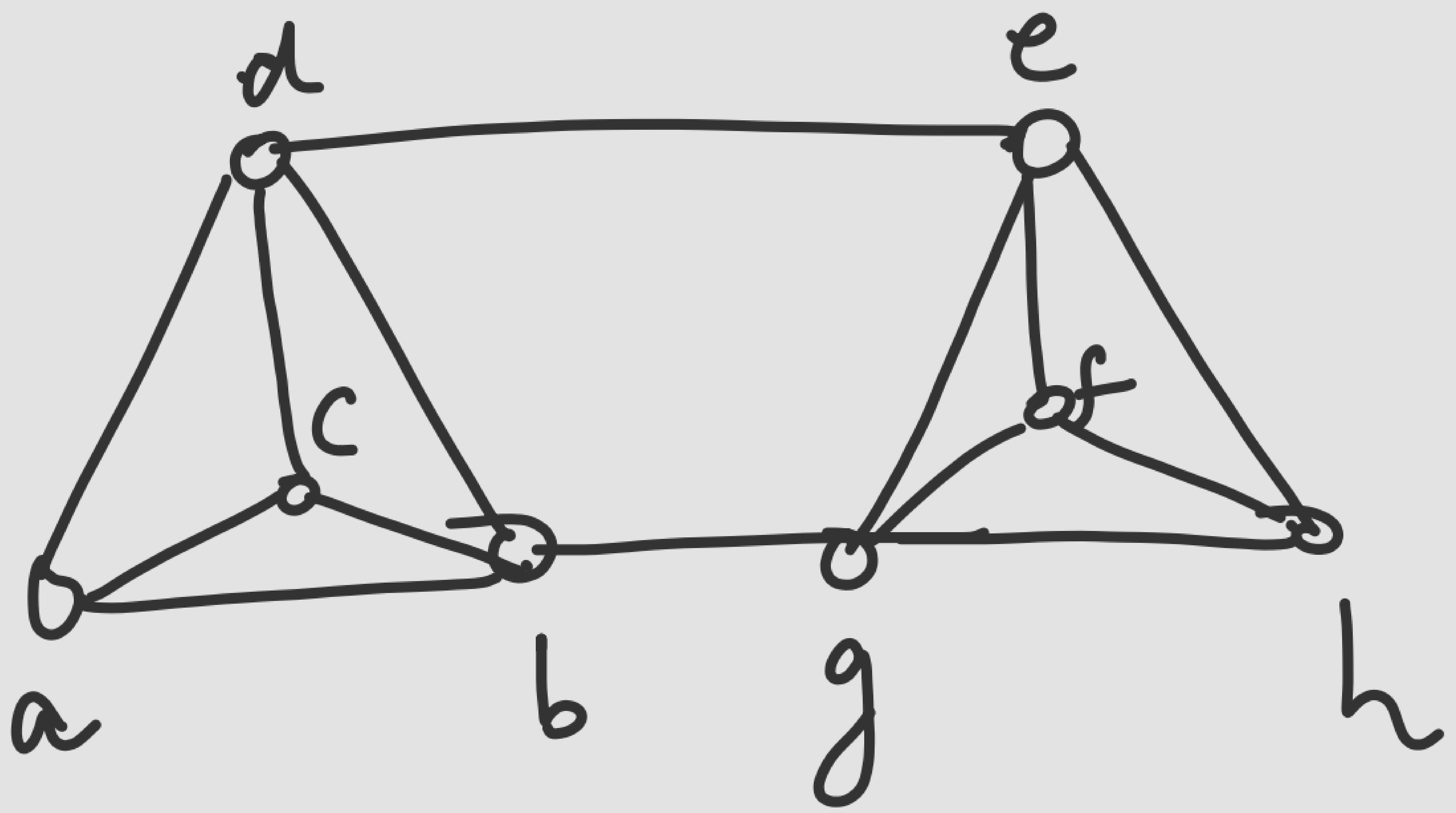


$G - \{c, i\}$



$K(G) = 2$

Compute $\delta(G)$, $\lambda(G)$ and $K(G)$



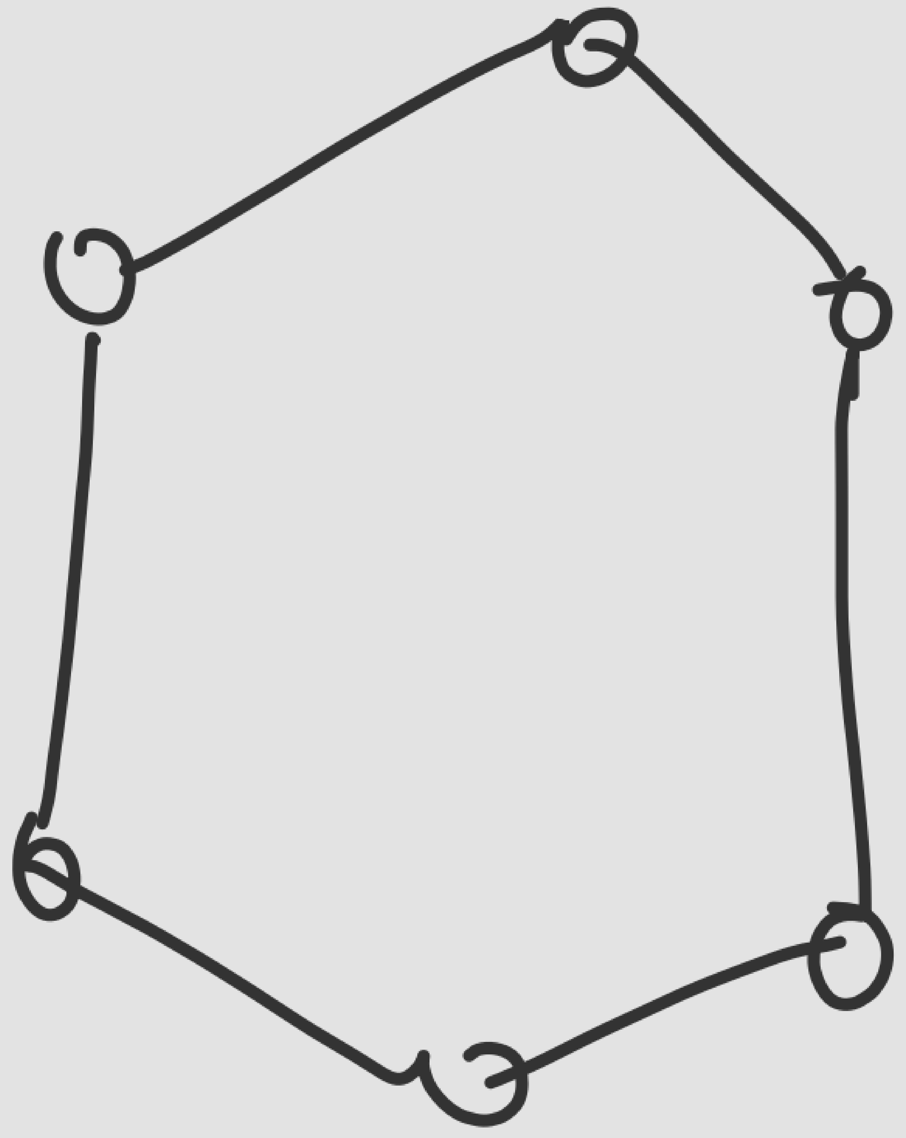
$\delta(G) = 3$

$\boxed{K(G) = 2}$

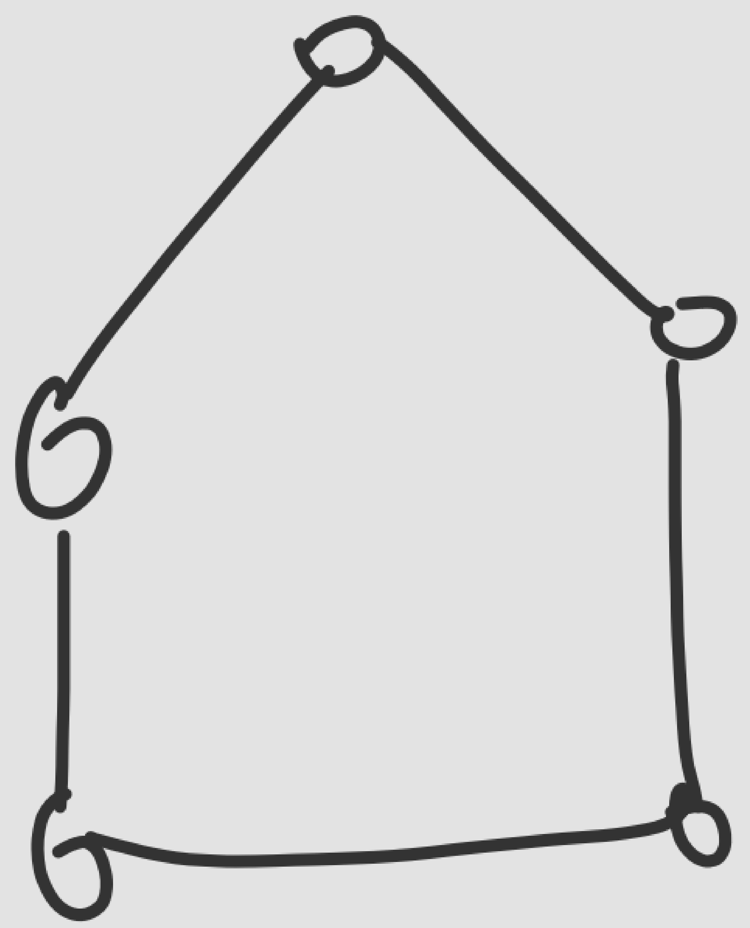
$\lambda(G) = 2$

$K(G) \le \lambda(G) \le \delta(G)$

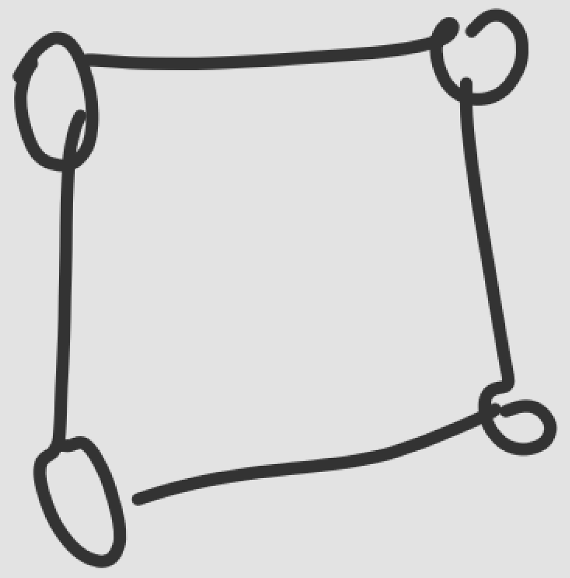
$\le \underline{2} \le 3$



$K(C_6) = 2$



$K(C_5) = 2$



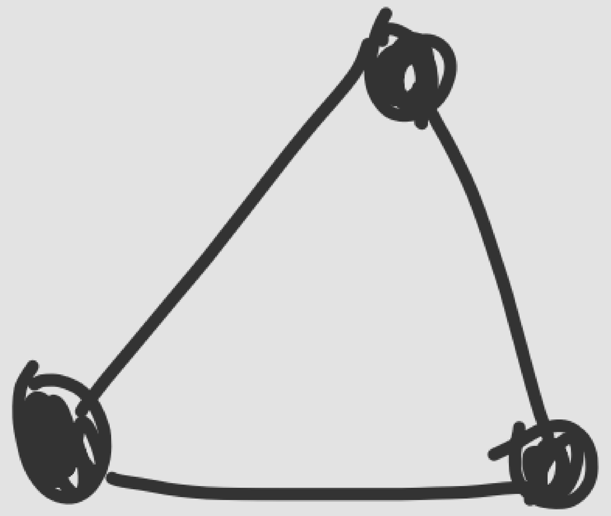
$K(C_4) = 2$

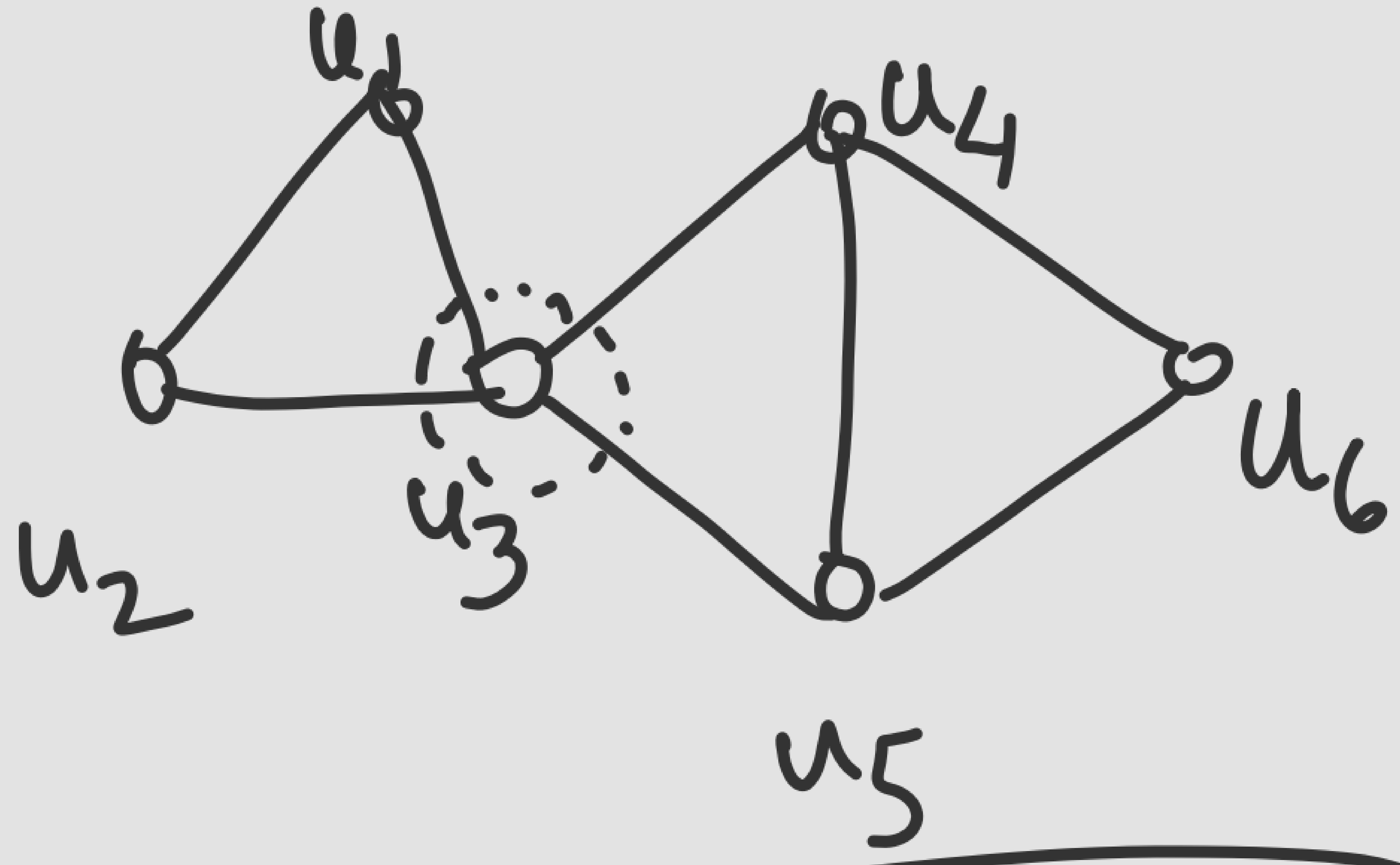
$\boxed{K(C_n) = 2}$

K- connected graphs

K(G)

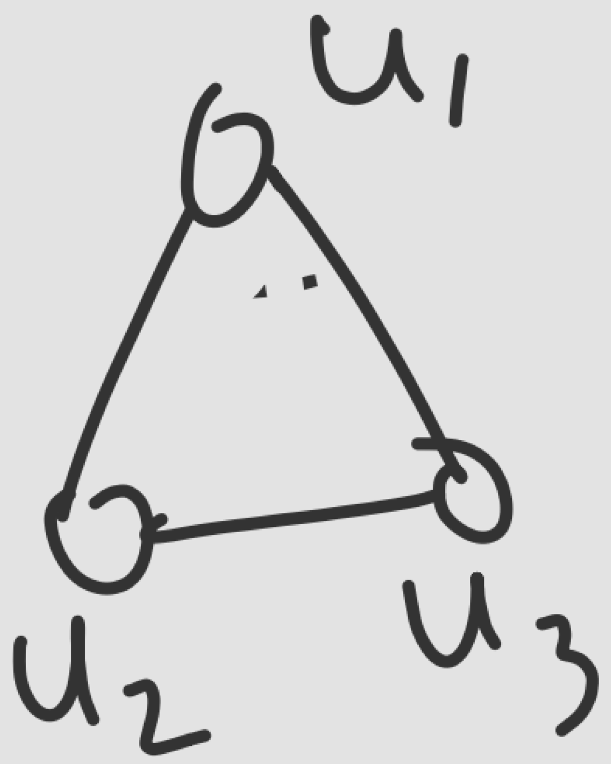
A graph is K-connected if and only if we cannot disconnect it by deleting fewer than K vertices.

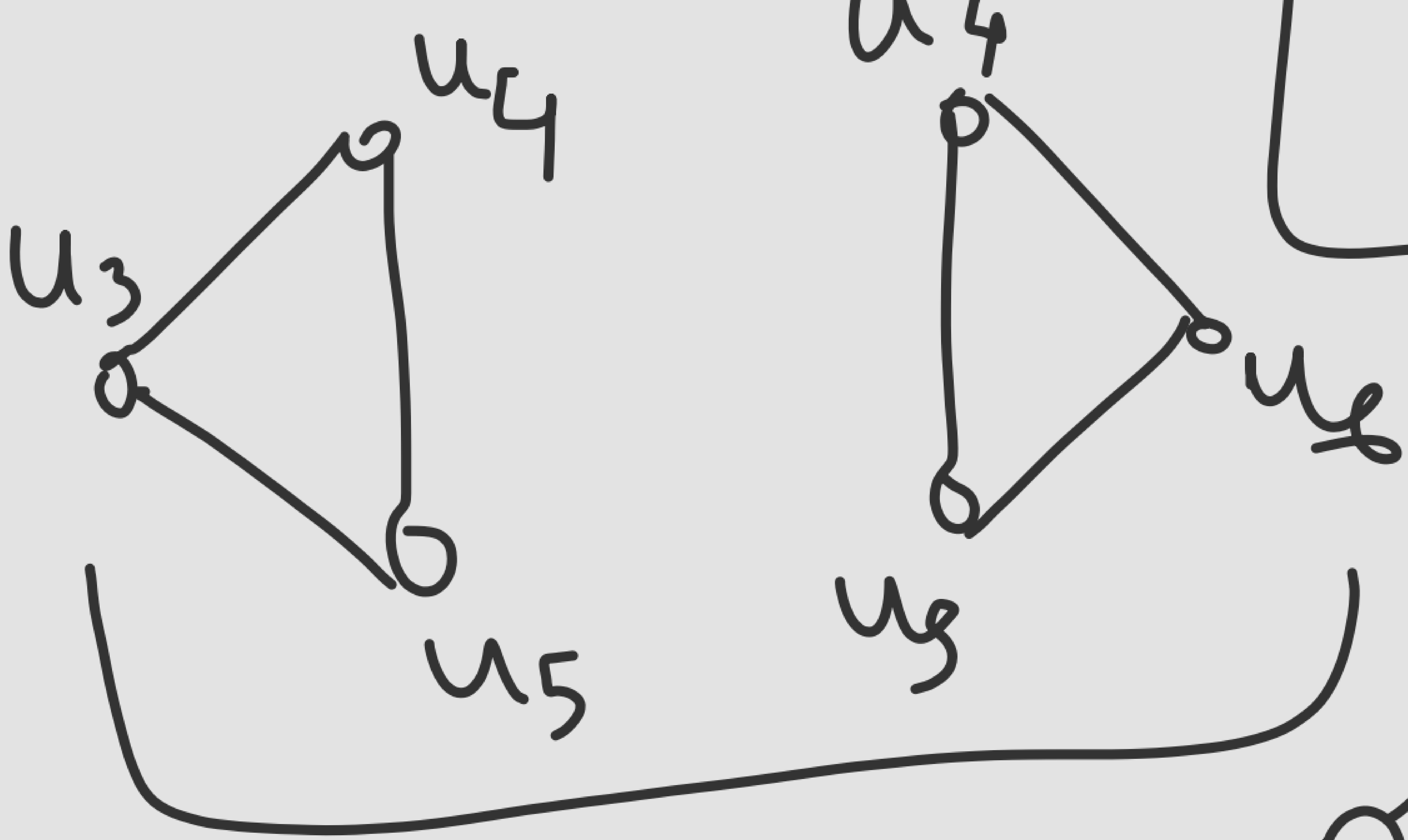




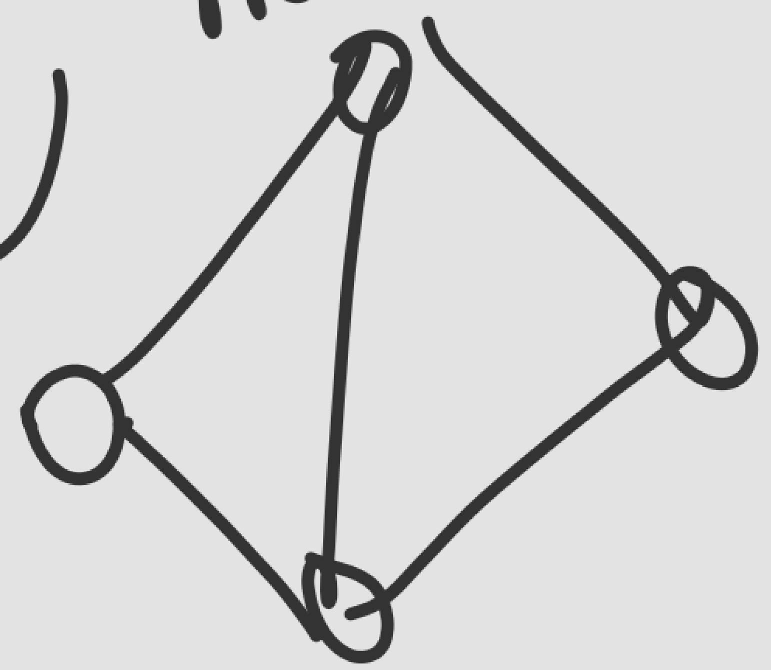
1. non-trivial
2. no cut vertex
3. connected

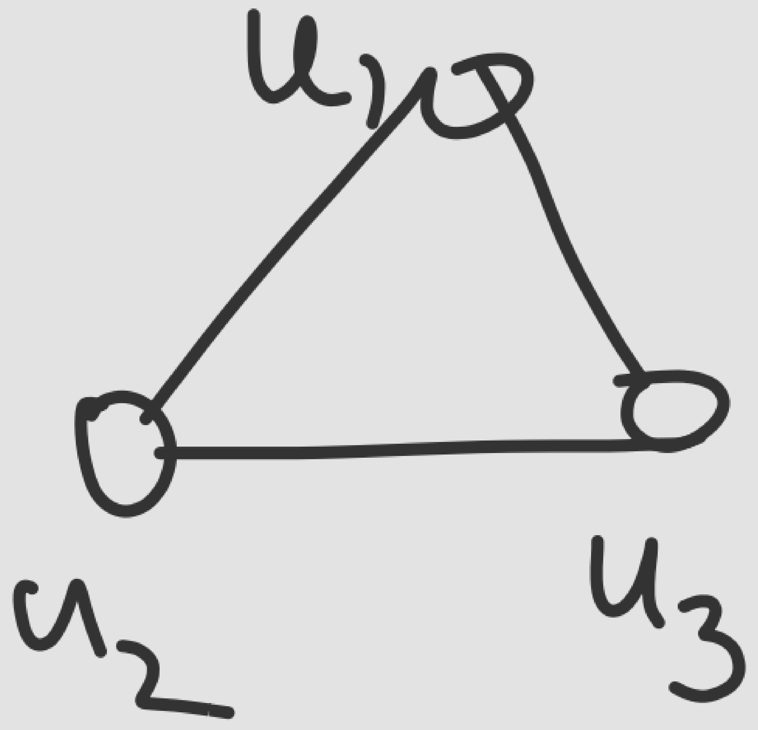
non-seperable graphs.

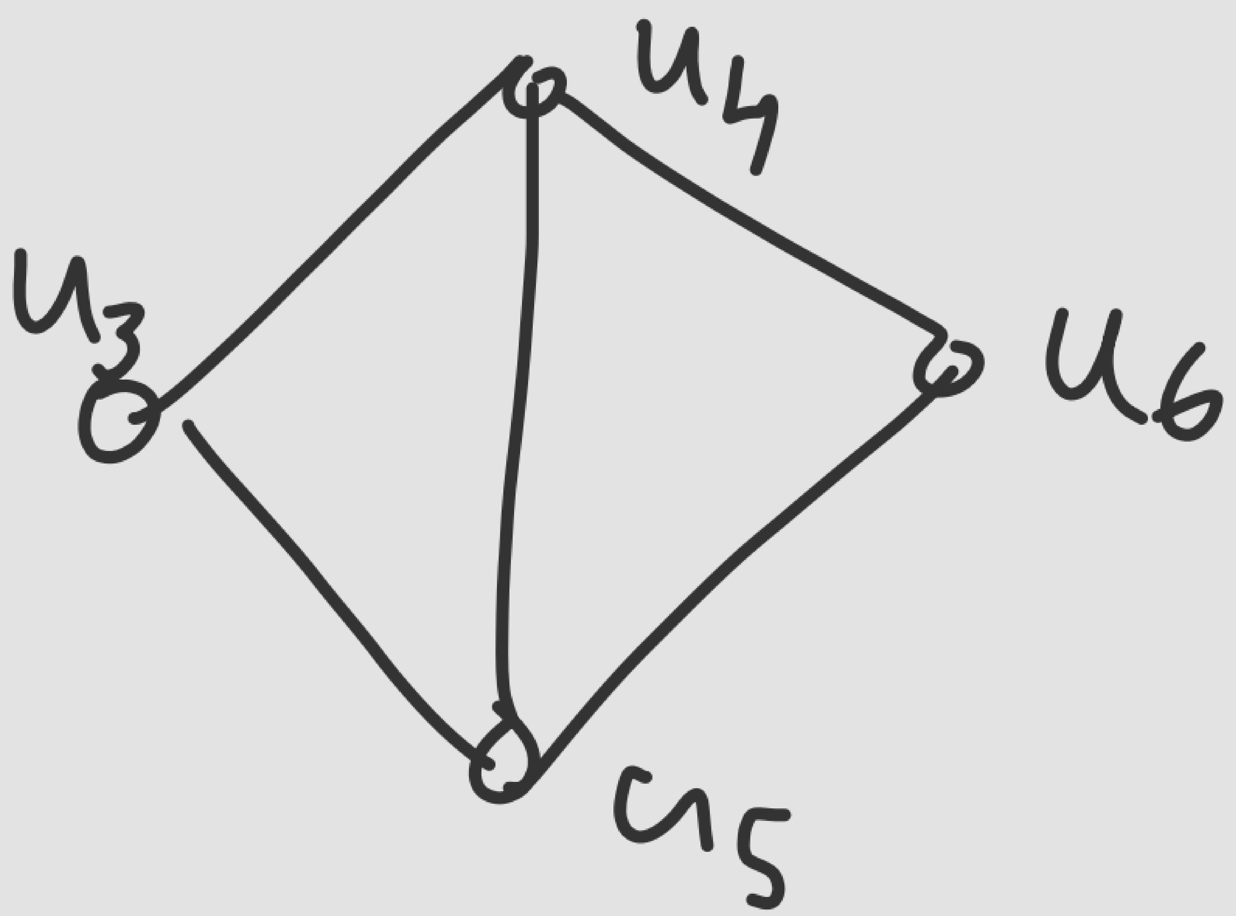




maximal







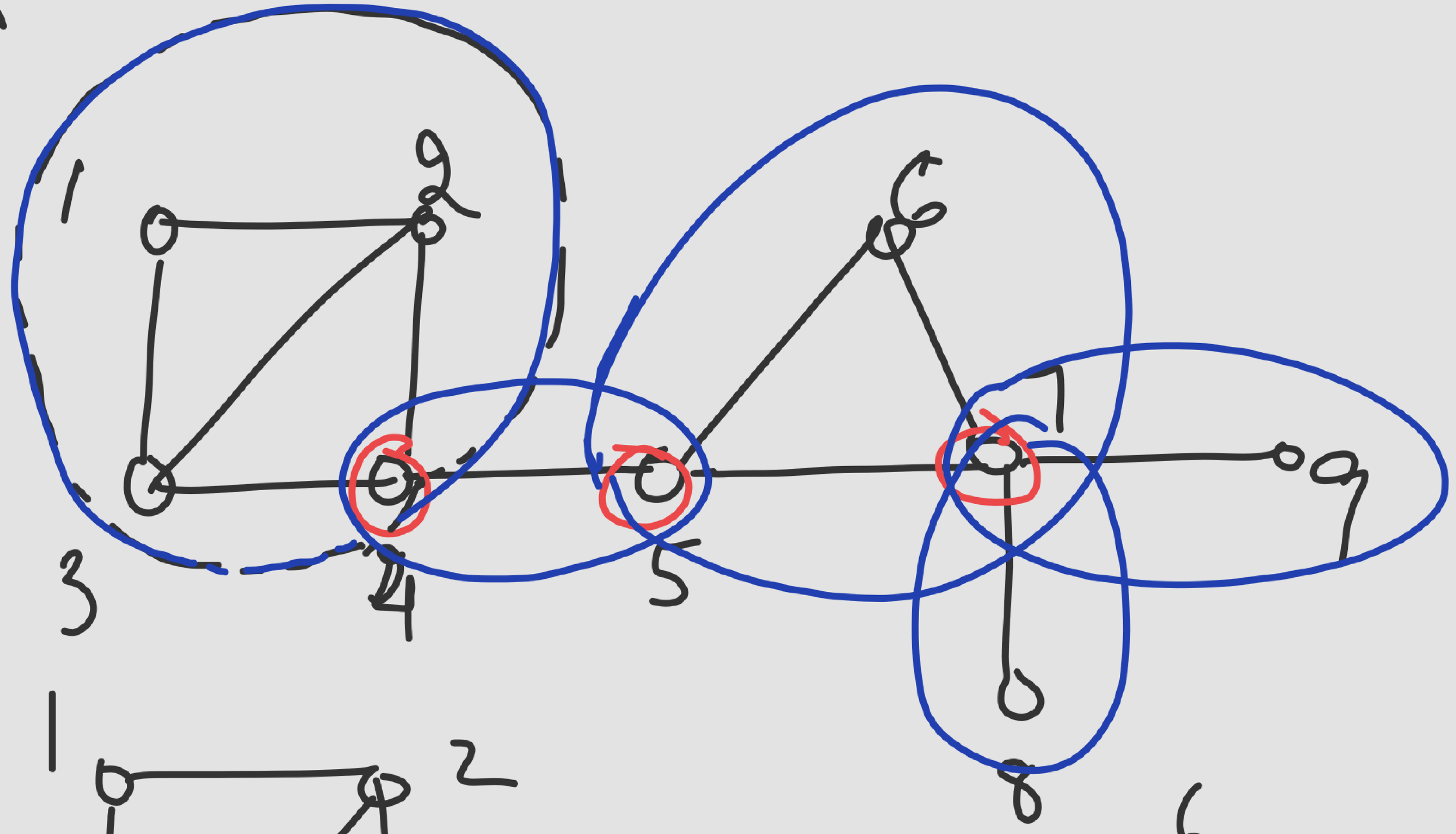
1. non-seperable
2. Maximal

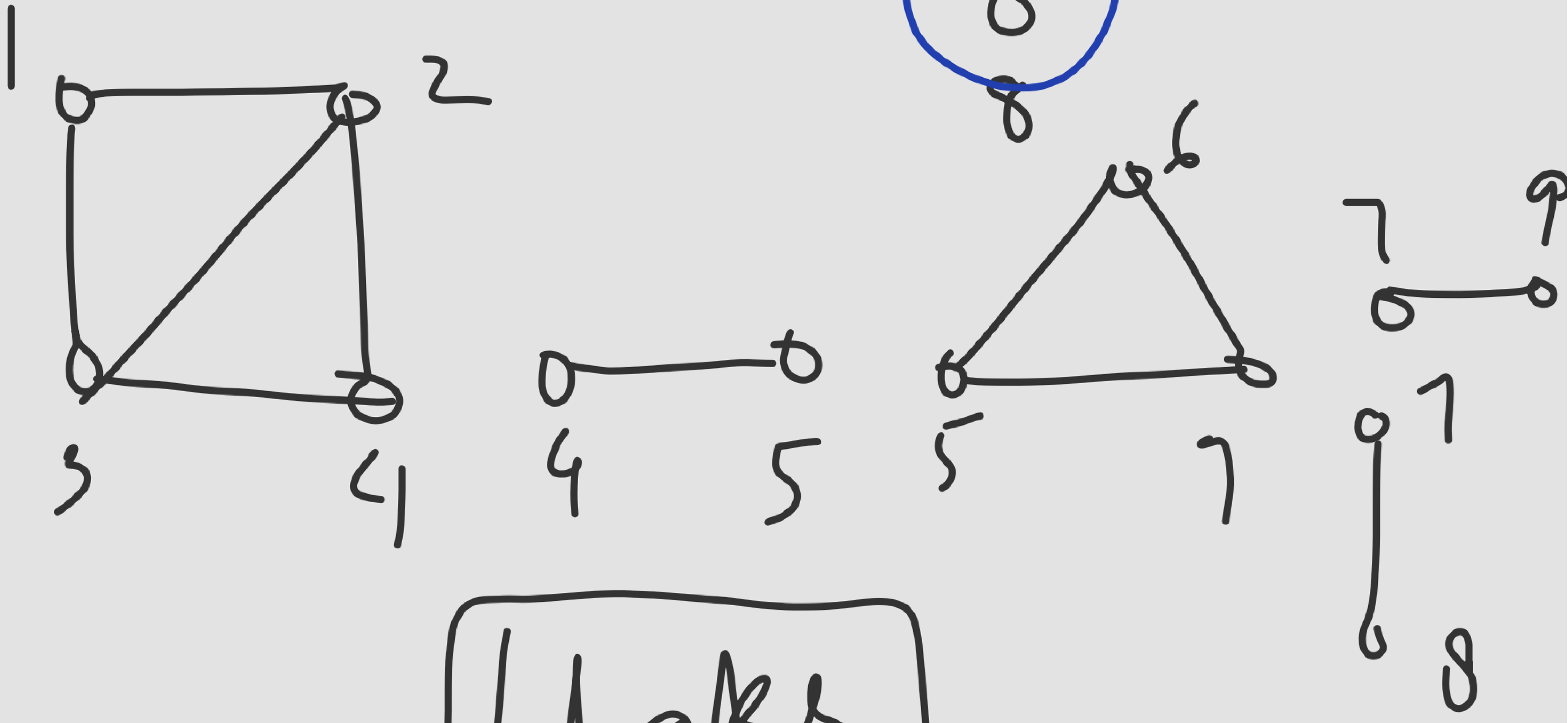
Blocks

maximal non-seperable subgraph of a graph.

endblock

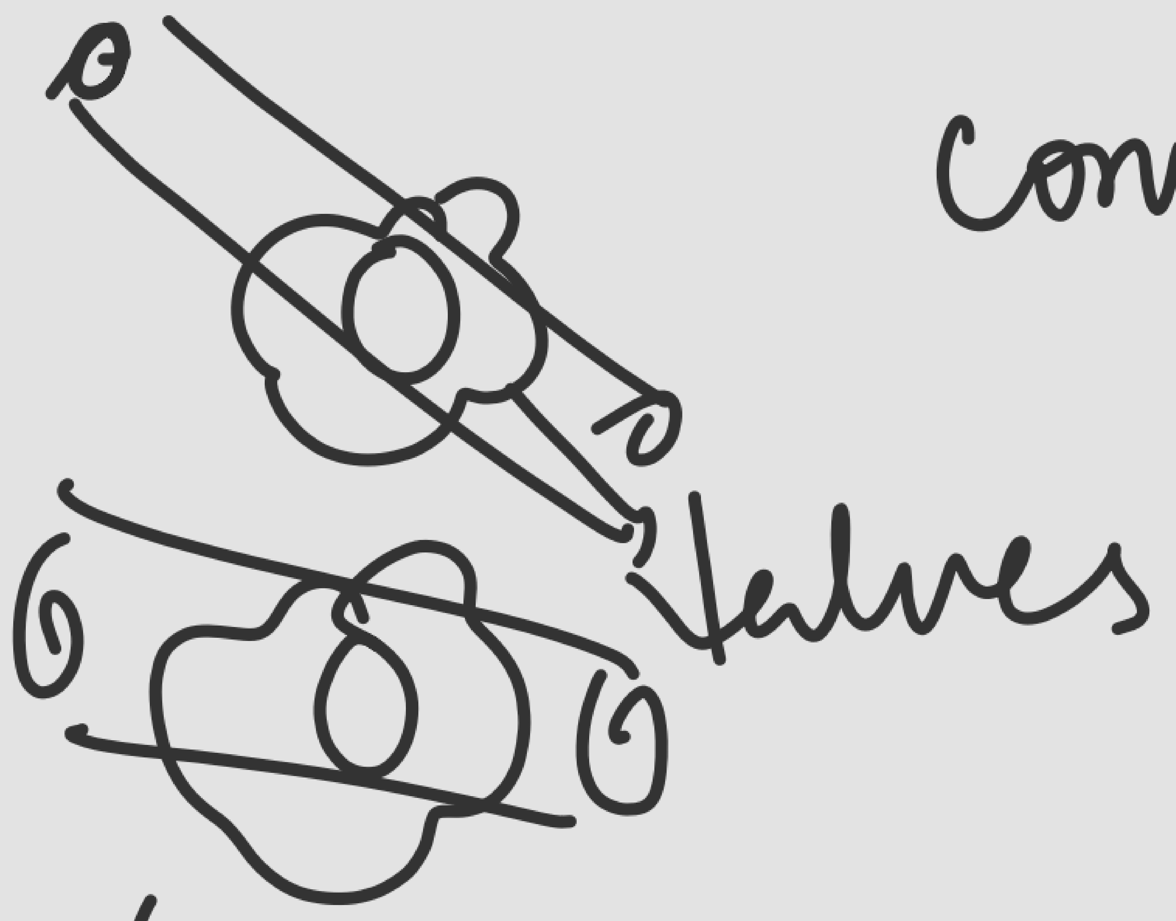
blocks which have one cut vertex



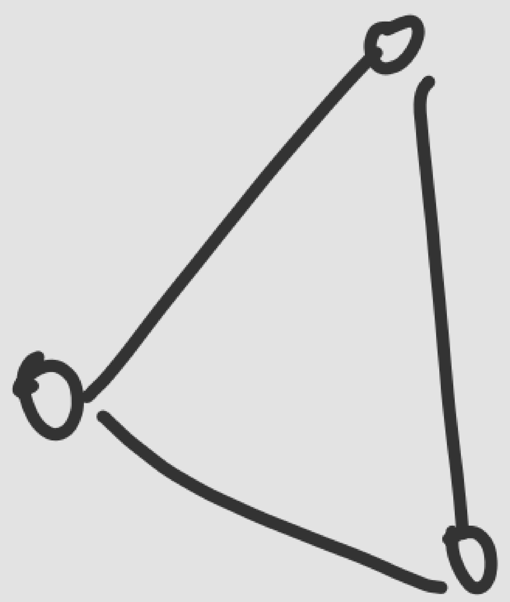


blocks

Network flow



graph



vertices – junctions

edges – pipes

weight/capacity – amount that flows

Network

A digraph which has non-negative capacity $c(e)$ on each edge e and two distinguished vertices called Source 's' and sink 't'.

Source s

The source vertex has only outward edges

Sink 't'

The sink vertex has only incoming edges

Capacity c(e)

non negative weight of an edge

Flow f(e)

for a flow to be feasible, the following must hold —

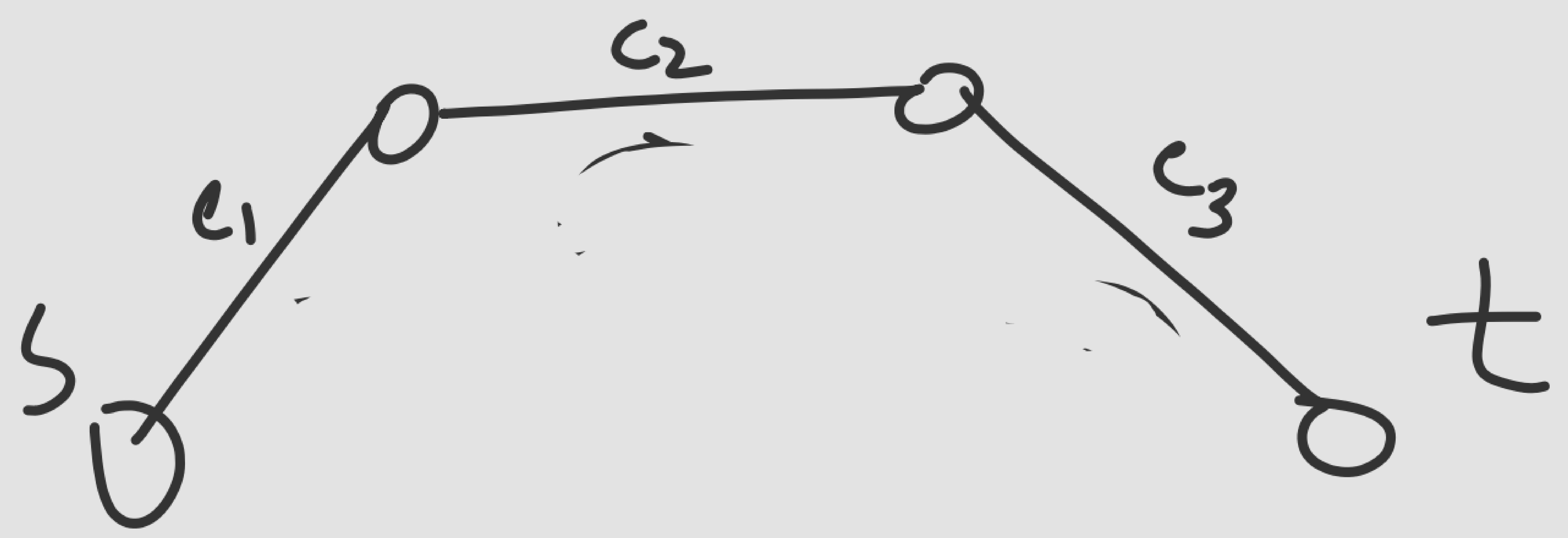
1) $f(e) \leq c(e)$ → capacity constraint

11) $f^{+}(e) = f^{-}(e)$ → conservation constraint

$f^{+}(v)$ → total flow on edges <u>leaving</u> v

$f^{-}(V)$ → total flow on edges entering V

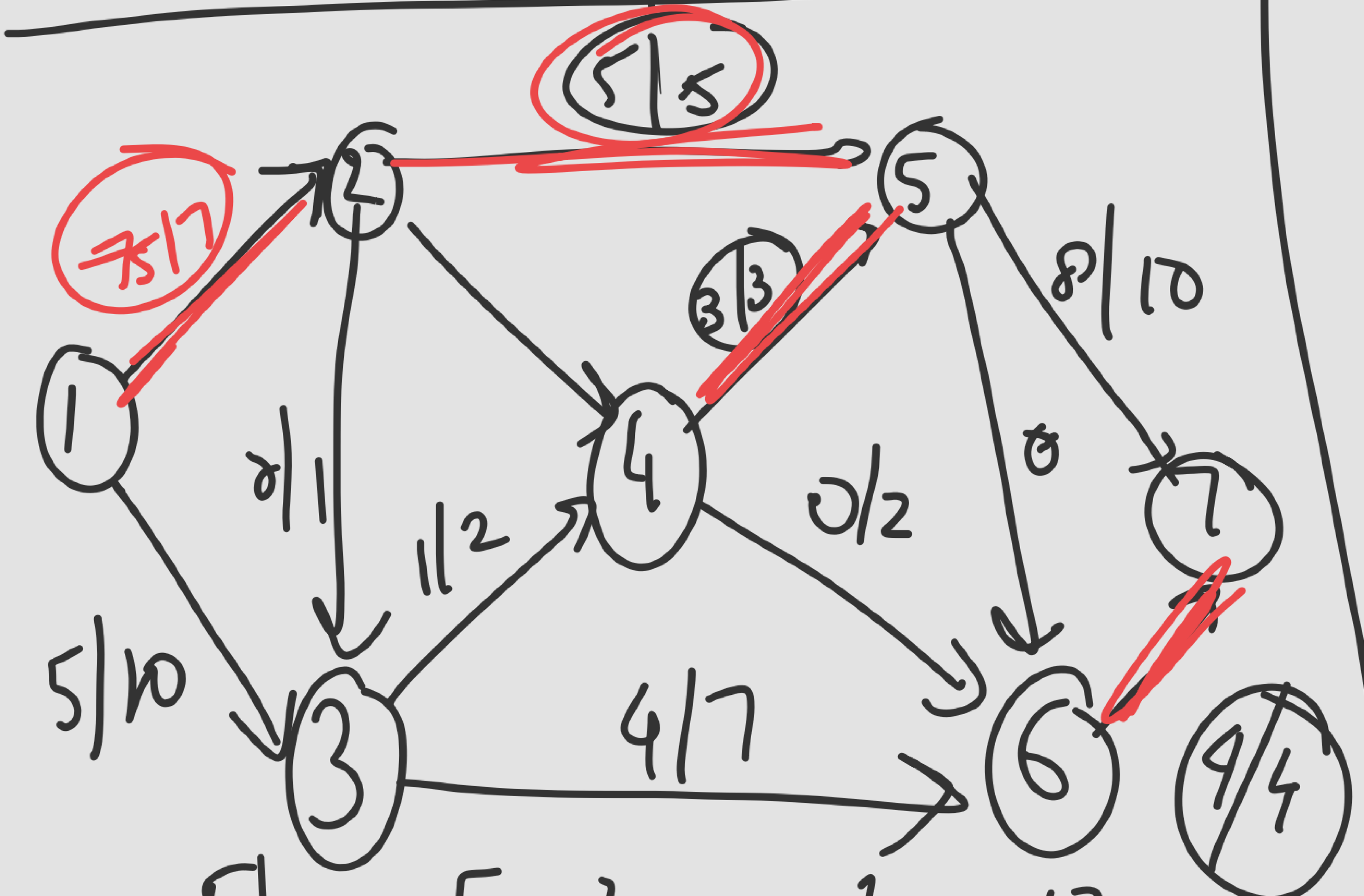
Bottleneck



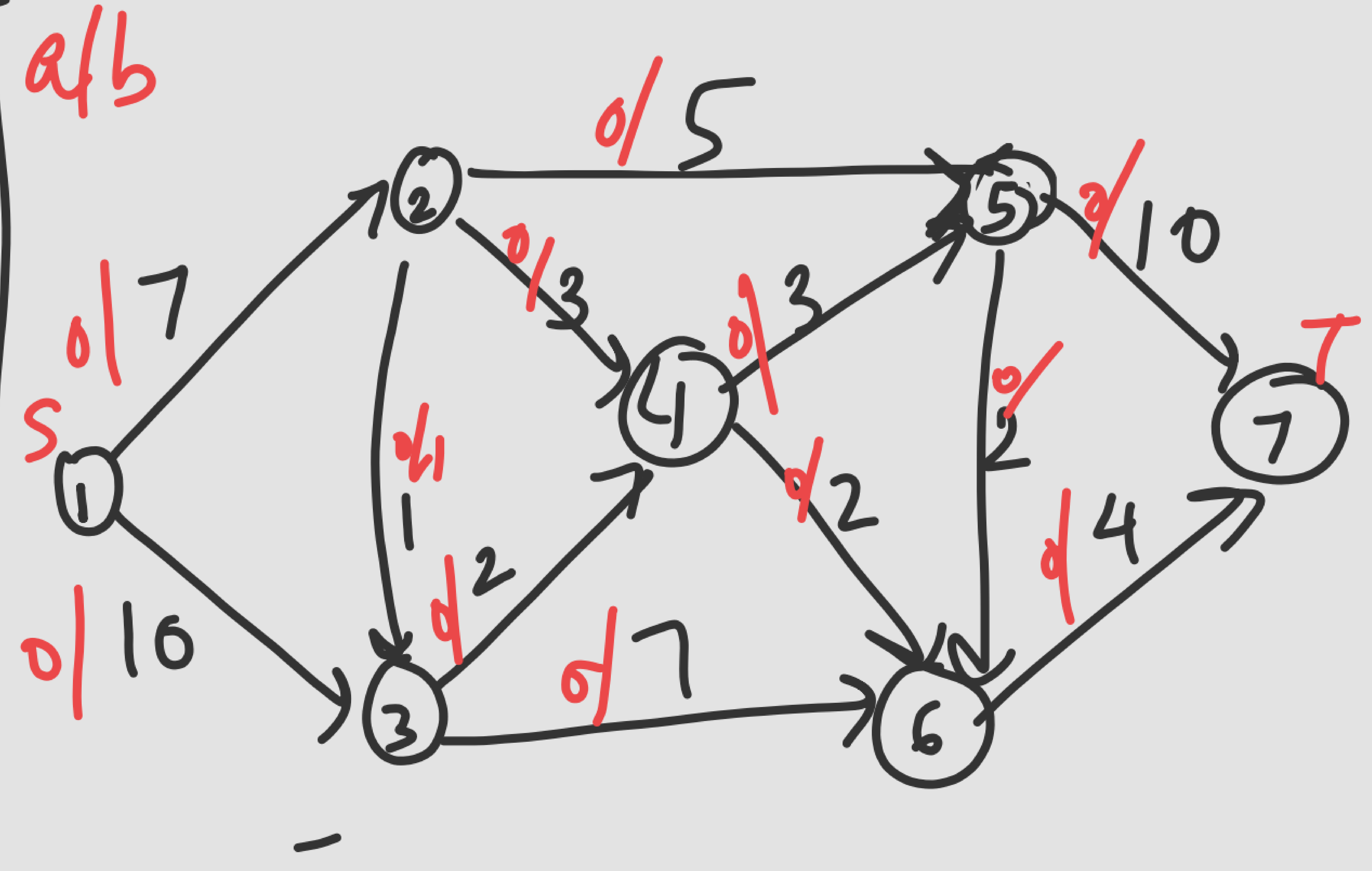
Along the path from s–t, an edge with minimum ~~tie~~ capacity is bottleneck

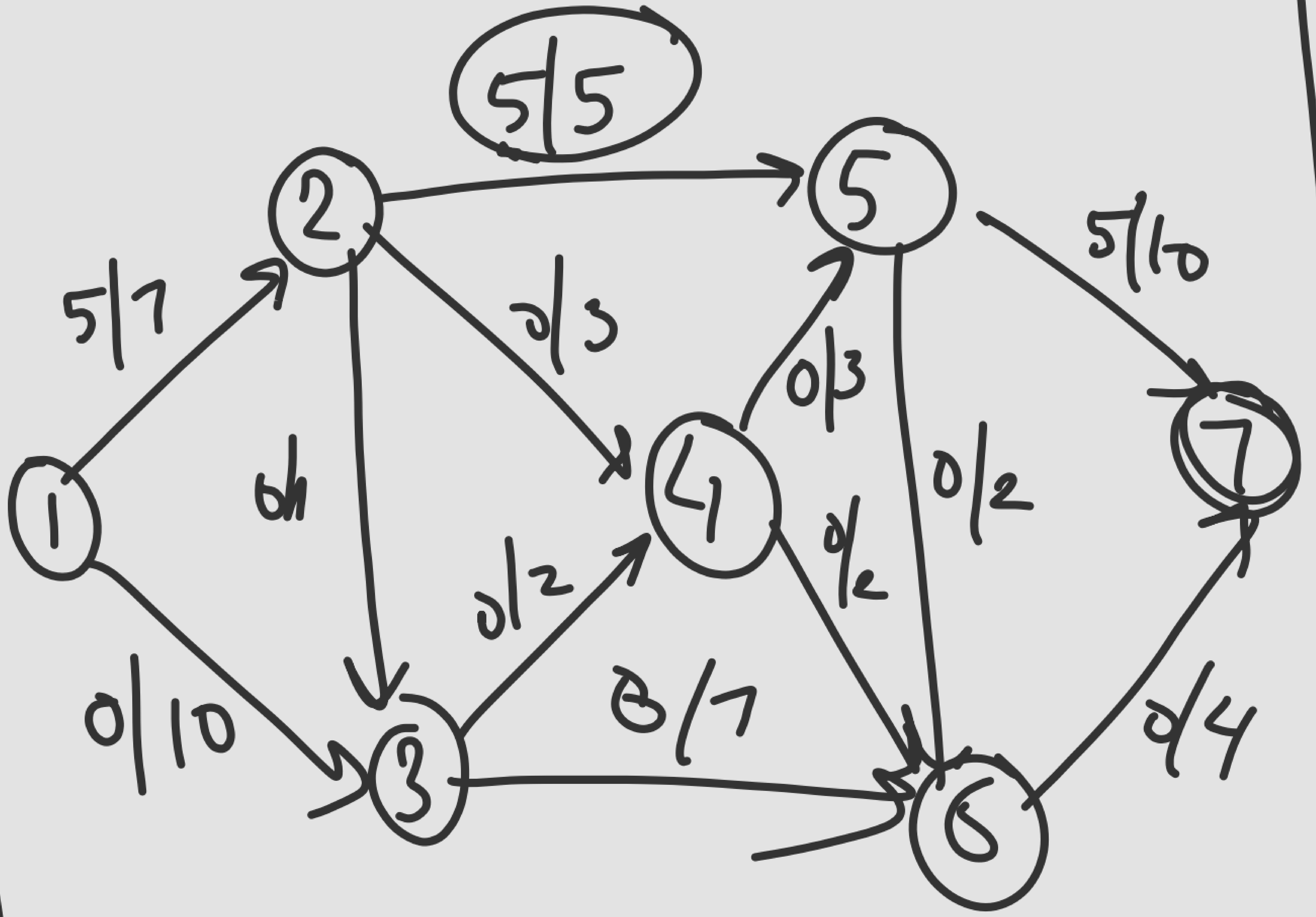
Augmenting path

Augmenting paths	Bottleneck
1-2-5-7	5
1-3-6-7	4
1-2-4-5-7	2
1-3-4-5-7	1

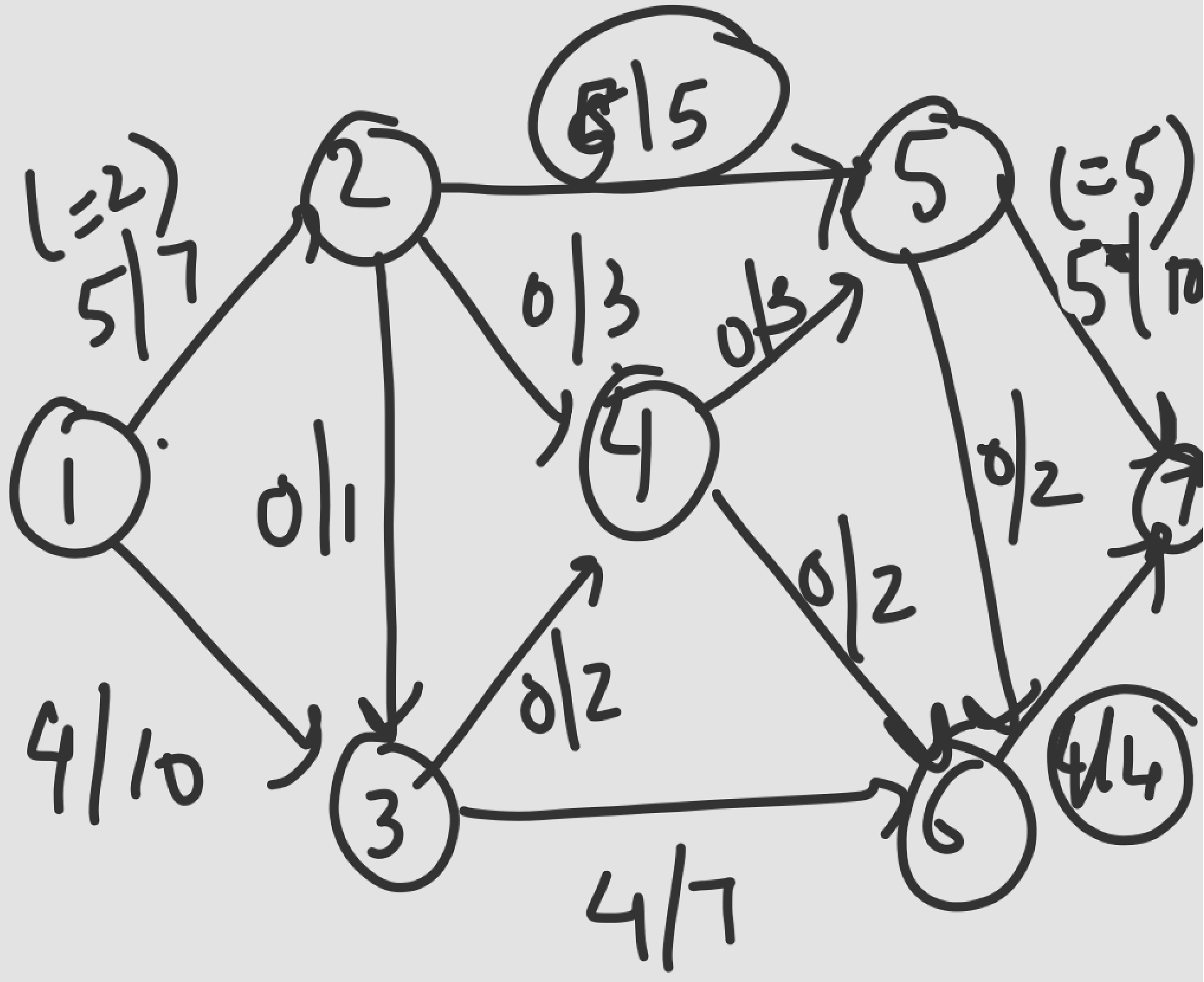


Flow = 5+4+2+1 = 12

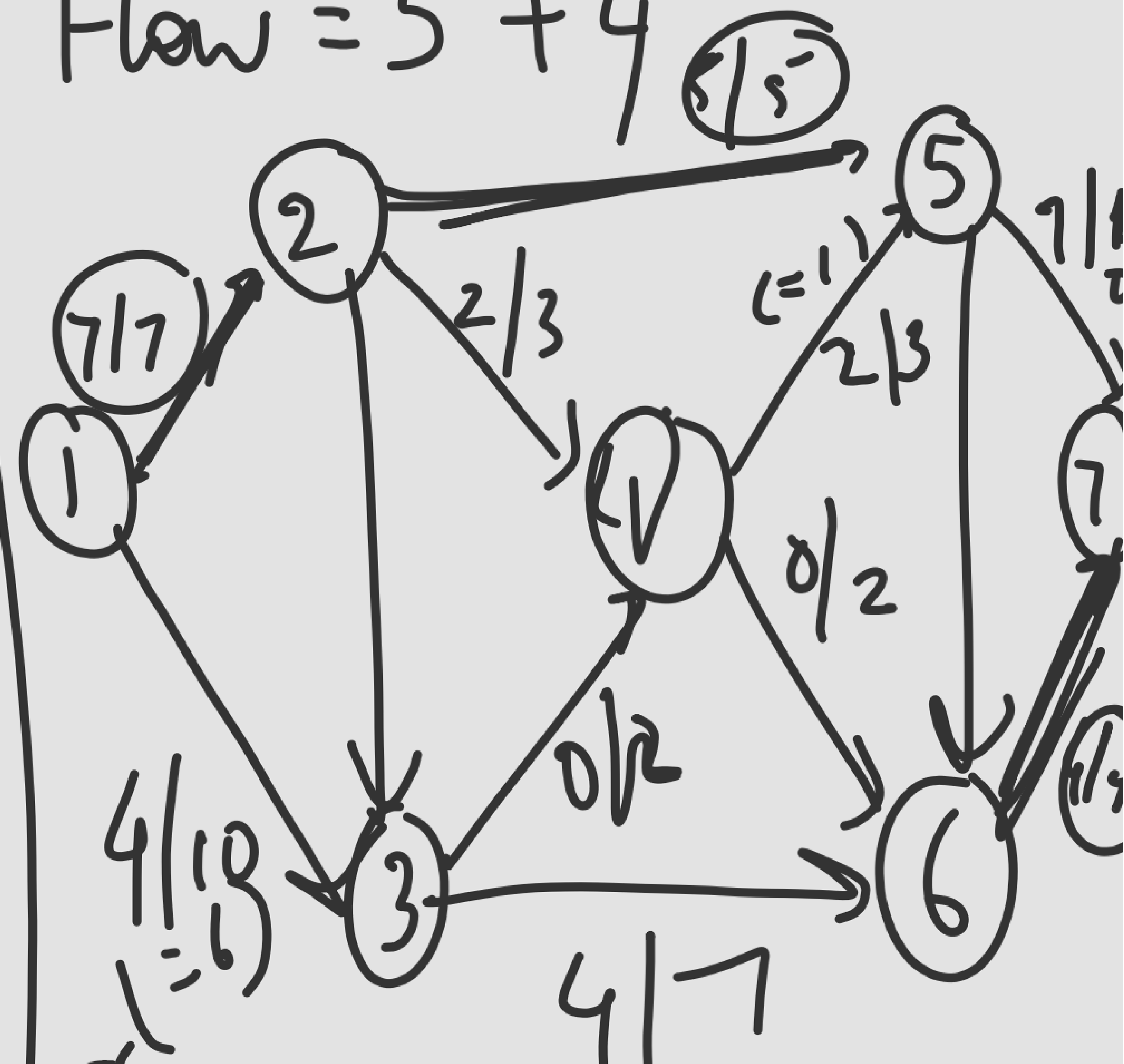




Flow = 5



Flow = 5 + 4



Flow = 5+4+2