edge/vertex connectivity
cut set
min cut
max flow
augmenting path
residual graph

SYSM 6302

CLASS 6

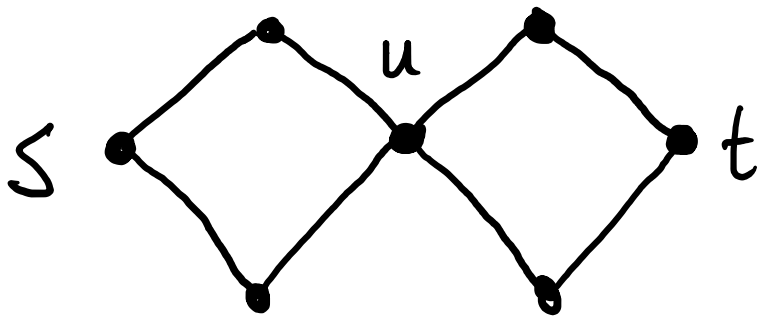
Connectivity between two nodes = # of **independent paths** between them

edge or vertex

edge or vertex

This statistic captures how strongly connected two nodes are

↳ Simultaneously identifies "bottlenecks" in the throughput from one node to the other



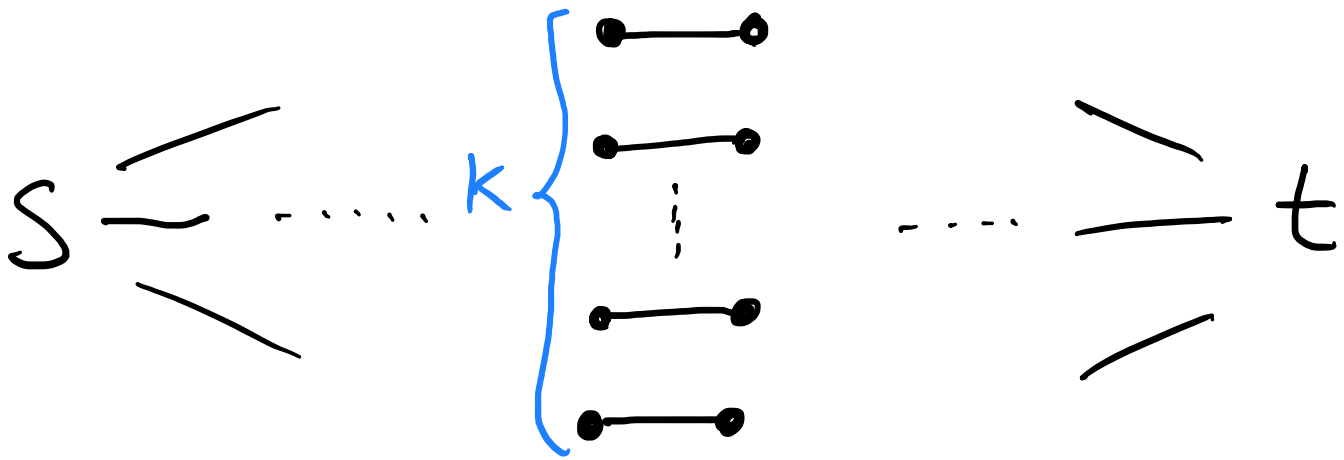
vertex/edge **cut set** is the set of vertices/edges that must be removed in order to disconnect two vertices

Size of the minimum vertex cut set = vertex connectivity = # of vertex independent paths

Size of the minimum edge cut set = edge connectivity = # of edge independent paths

Flow

How much material can be pushed through the network from s to t?



MAX FLOW
=
MIN EDGE CUT SET SIZE
=
# of EDGE INDEP PATHS

↑ edge cut set for s & t = # of edge independent paths from s to t

k indep paths ⇒ At least k × rate flow from s to t ≤ Max flow ≤ if we remove the k edges in the cutset, then flow is reduced by k × rate down to zero (s & t disconnected)

Max Flow - Min Cut

→ Applied without modification for directed networks

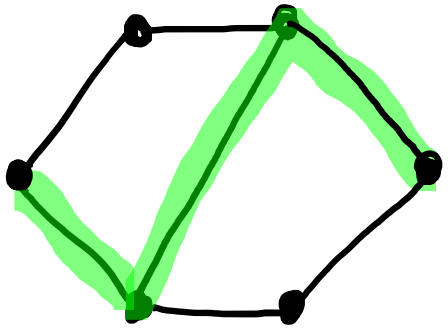
→ For weighted networks:

↳ Min cut is the cut set which has the minimum sum of edge weights

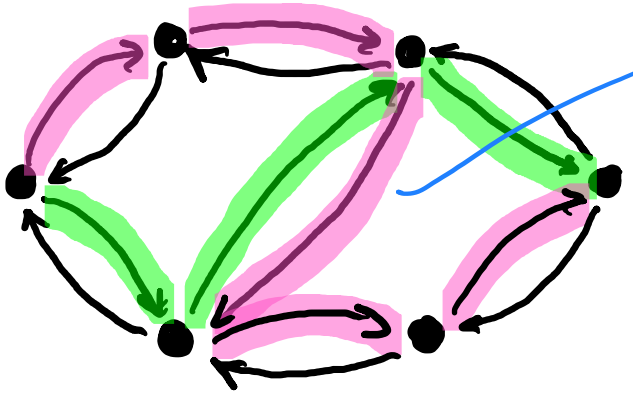
Ford-Fulkerson or Augmenting Path Algorithm

We consider the unweighted case for clarity

```
Augmenting Path (G, s, t):
    paths = ∅
    do
        p = BFS(G, s, t)
        G = G / p
        paths = paths ∪ p
    while p ≠ ∅
```

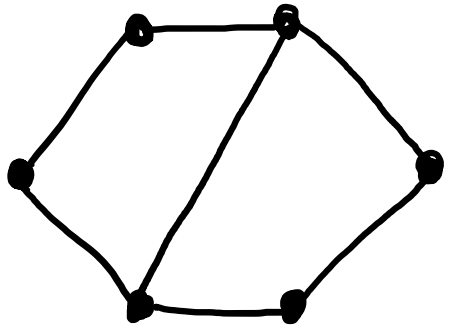


There should be two edge independent paths but this BFS result hides it!

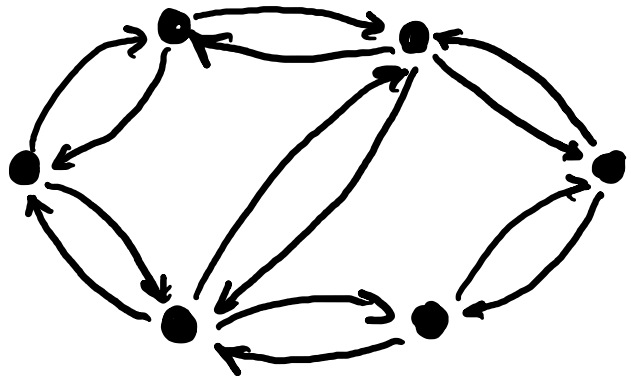


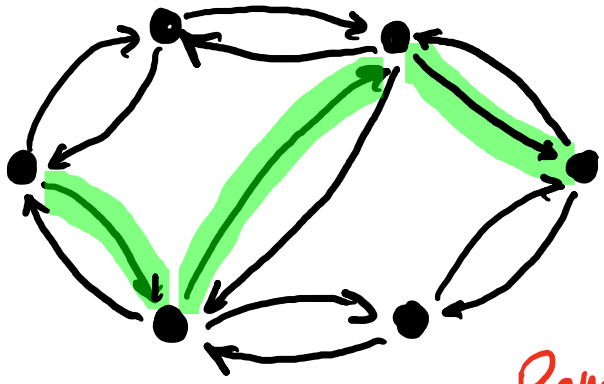
Net zero flow on this edge, which is what we expect!

ORIGINAL GRAPH



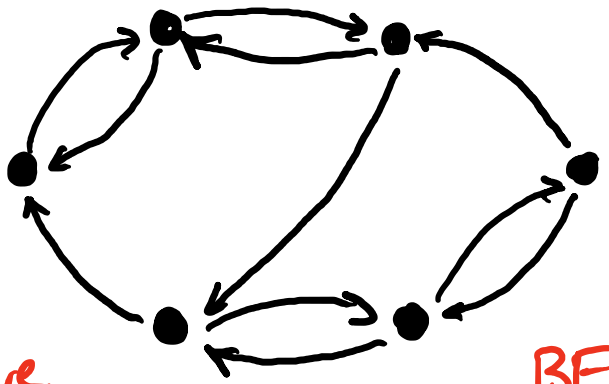
RESIDUAL GRAPH

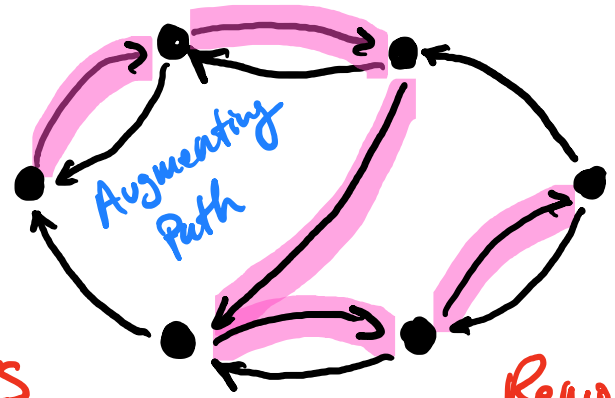




BFS

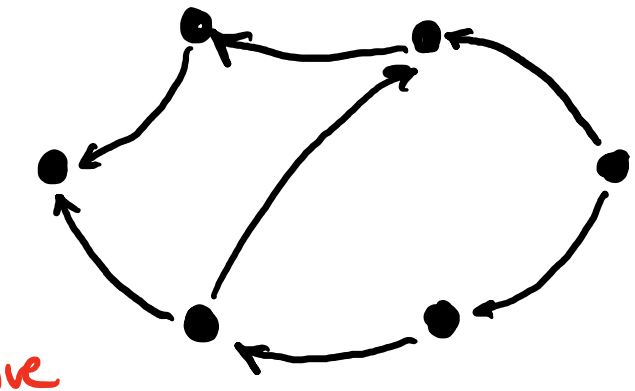
Remove edges

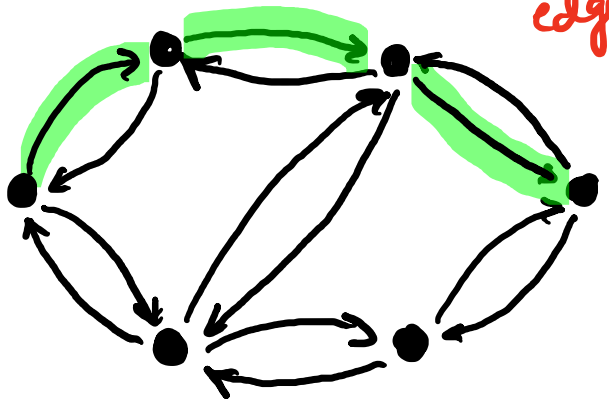


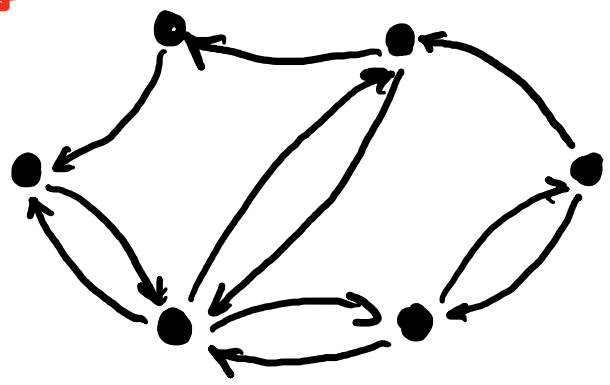


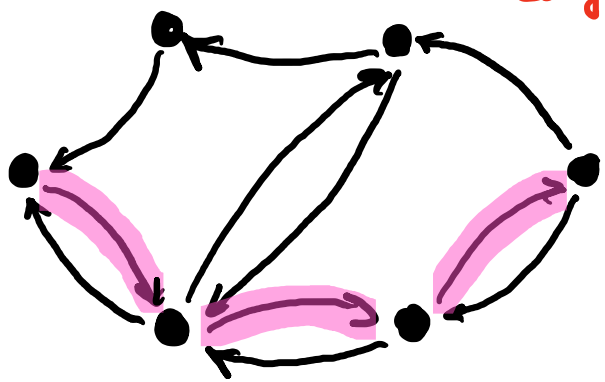
BFS

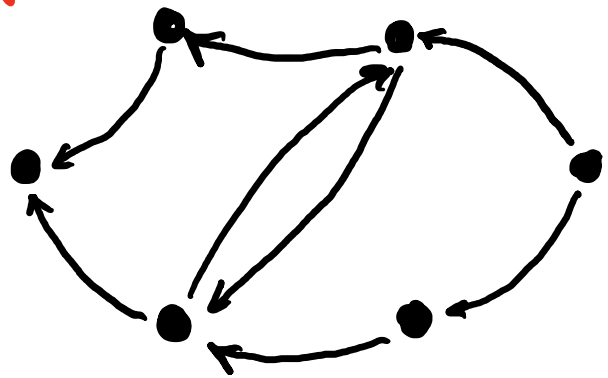
Remove edges



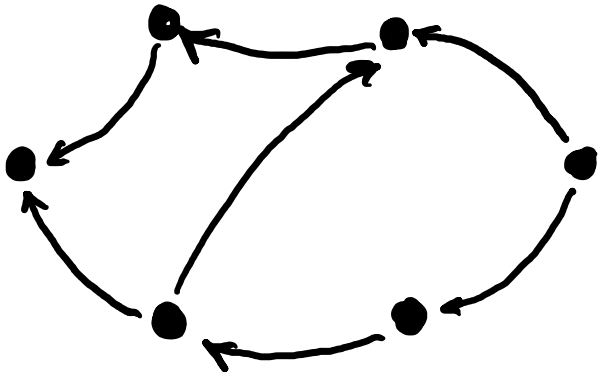








Final Residual Graph

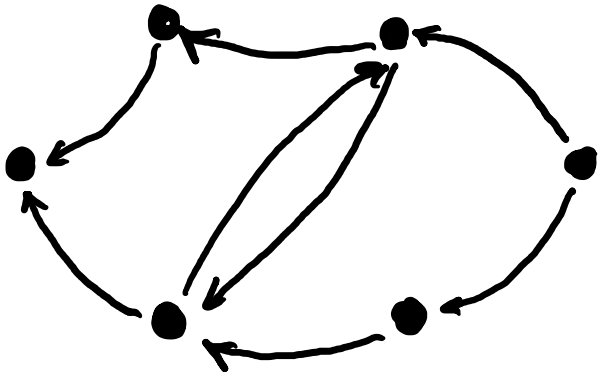


Find cut set (just unique!)

Find nodes reachable from $t = V_t$

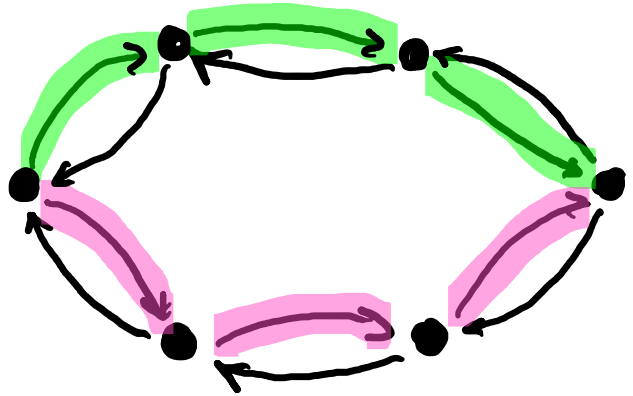
Find nodes reacable from $s = V_s$

→ cut set = edges that connect nodes in V_t with nodes in V_s

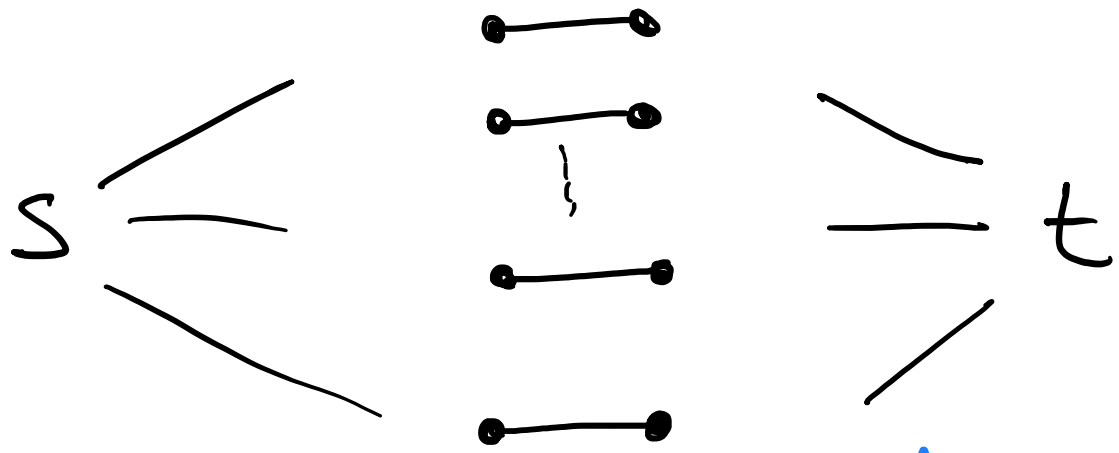


Find indep paths

Remove edges that go both ways between two nodes → BFS backwards



Running Time



↑ # of edge indep. paths is upper bounded by the degrees of s & t ↑

recall: BFS is $O(m+n)$

$$O(k(m+n))$$

k: # of edge indep paths

$(m+n)$: BFS

$$k \leq \min(\text{degree of } s, \text{degree of } t)$$

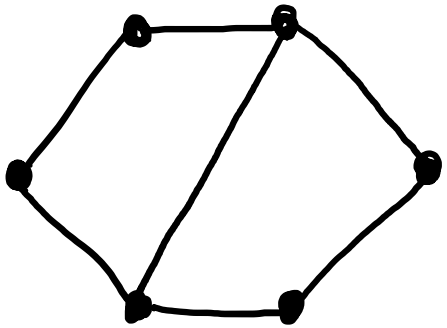
on average:

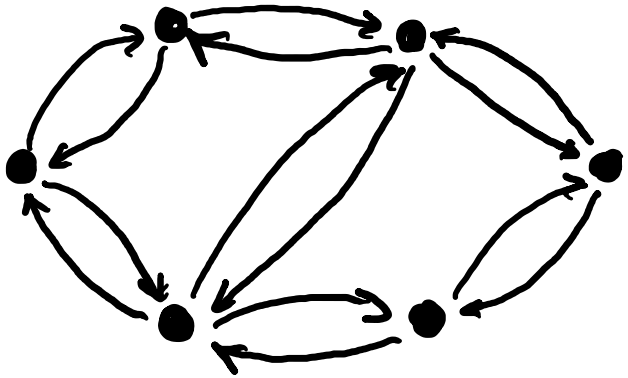
$$k \leq \frac{2m}{n}$$

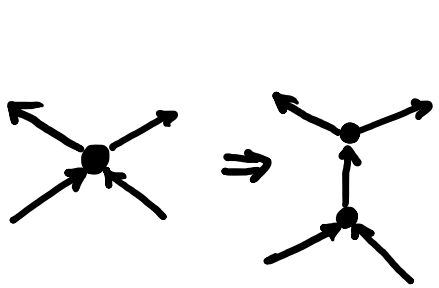
$$\Rightarrow O\left(\frac{m}{n}(m+n)\right)$$

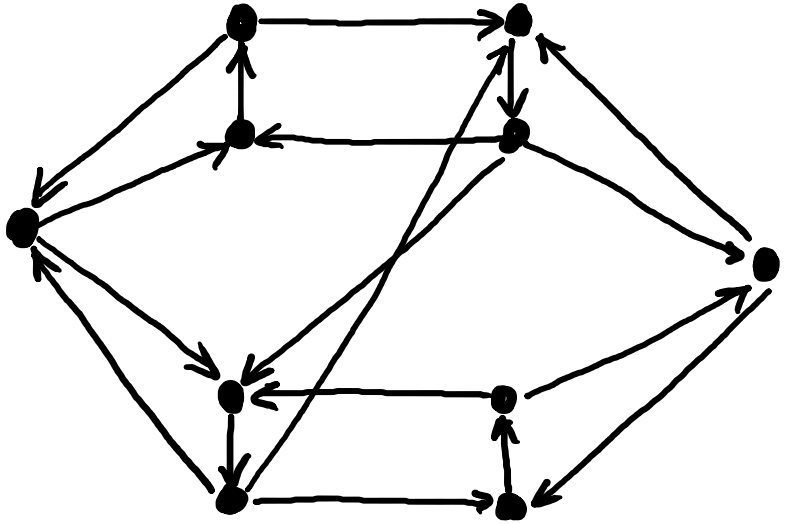
Vertex Independent Paths

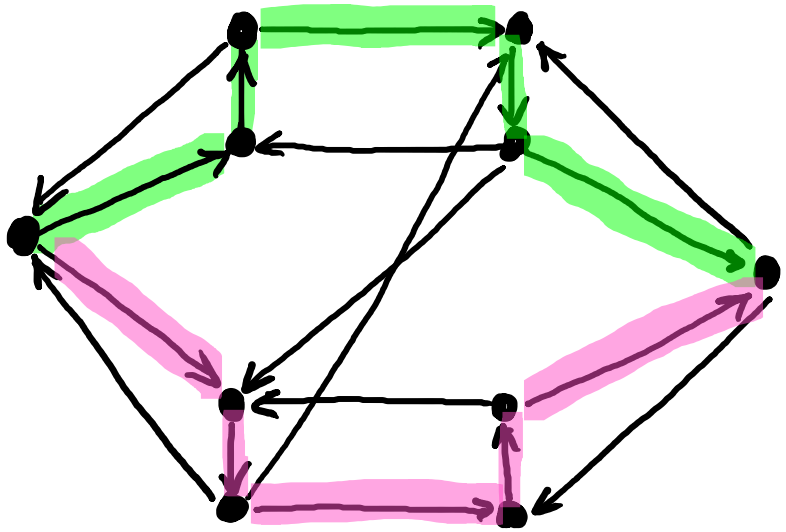
Edge independent paths and paths don't share vertices

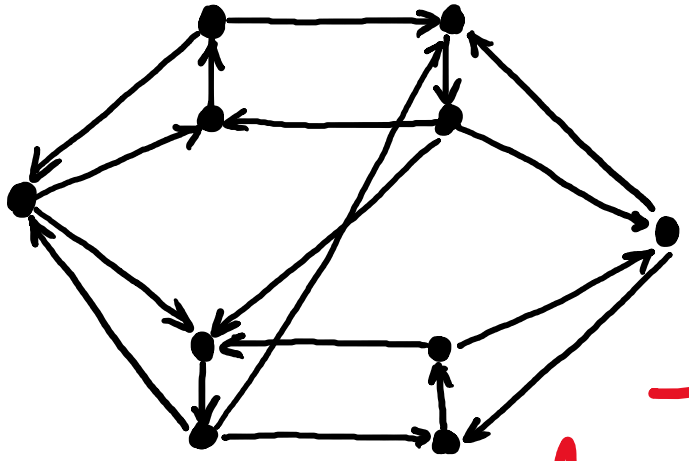


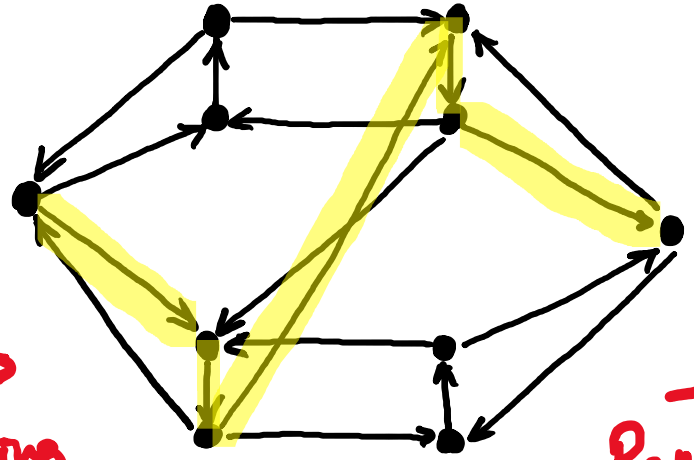




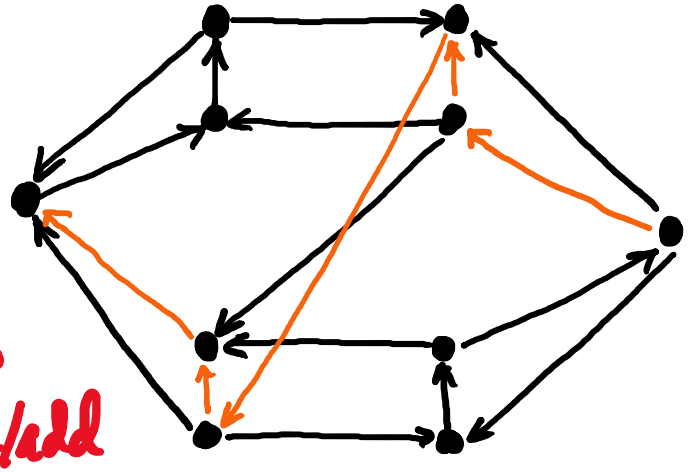




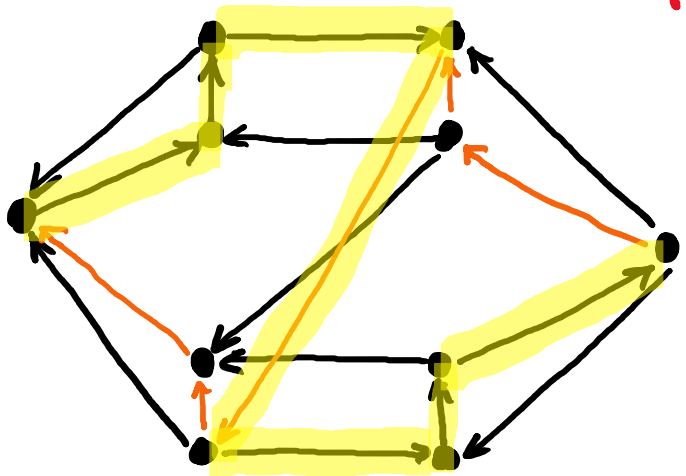




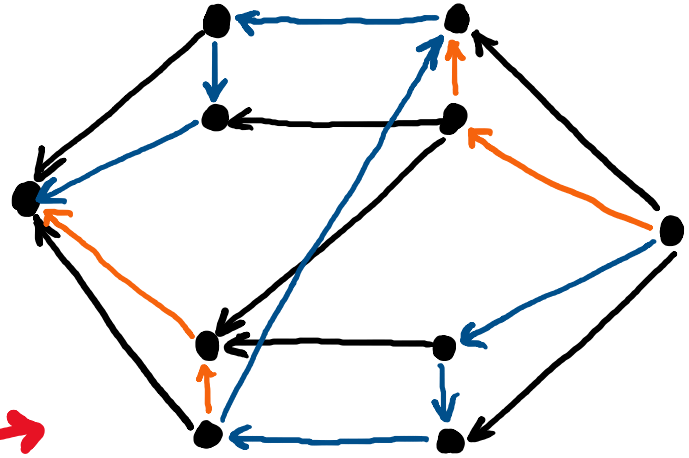
Augmenting Path



Remove/add edges



Augmenting Path



Remove/add edges