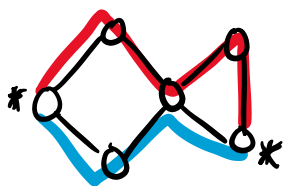


Find 27 paths
between marked
nodes that do
not share an edge

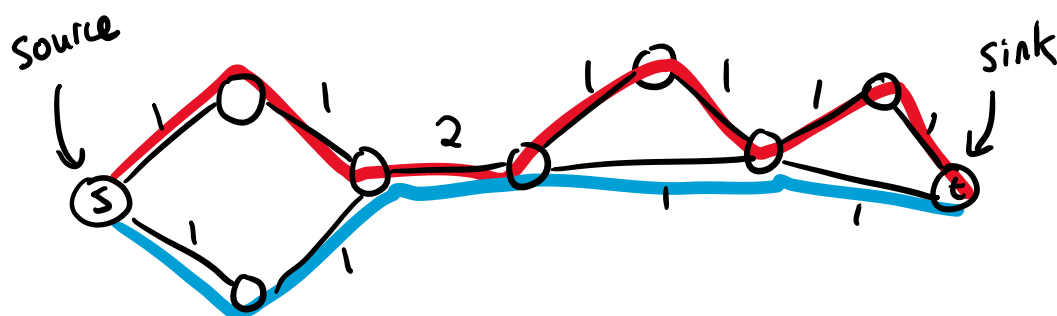


2ECC

cut bridges

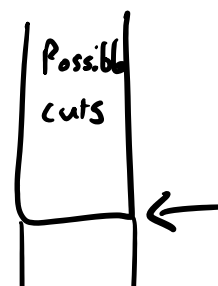
$O(V+E)$

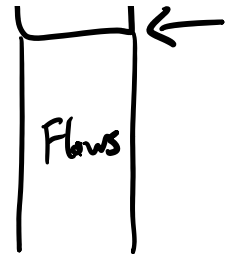
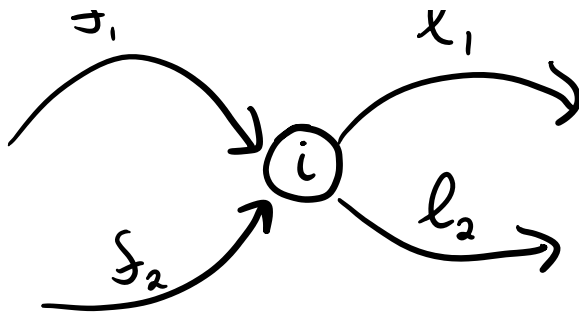
Edge Disjoint Path Problem (Max Flow)



Max Flow / Min Cut

Flow Conservation



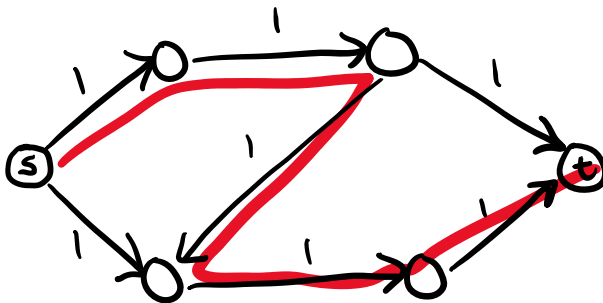


$$f_1 + f_2 = l_1 + l_2$$

except

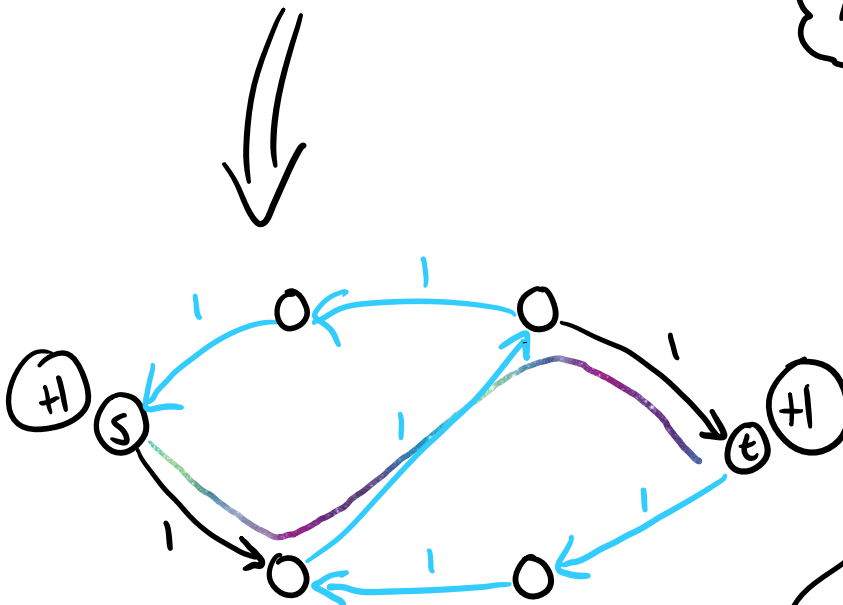
$$\textcircled{s} \quad \textcircled{t}$$

outflow = inflow



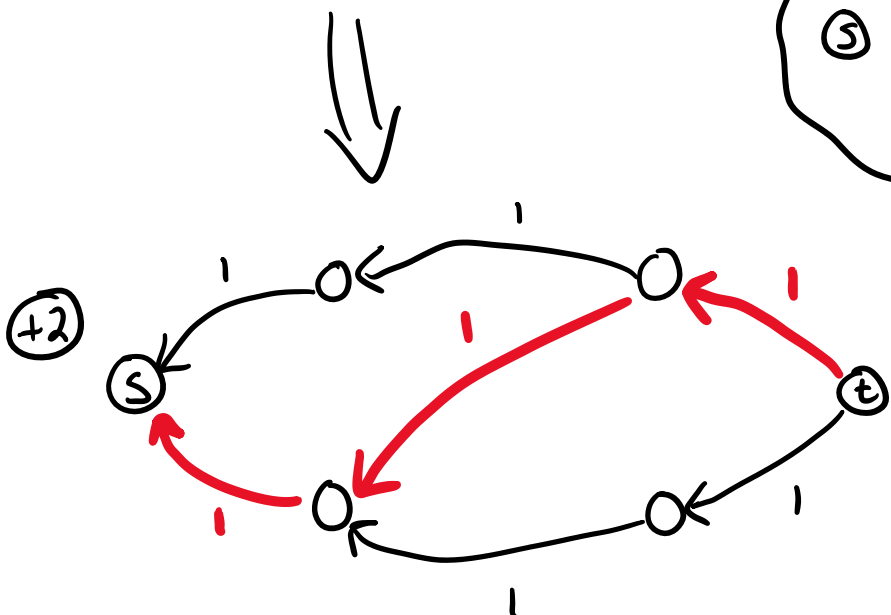
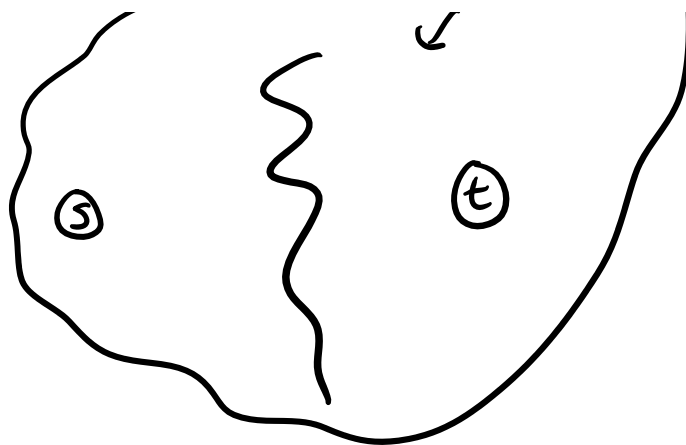
Wants
- be able to undo
a bad choice!

Augmenting Paths

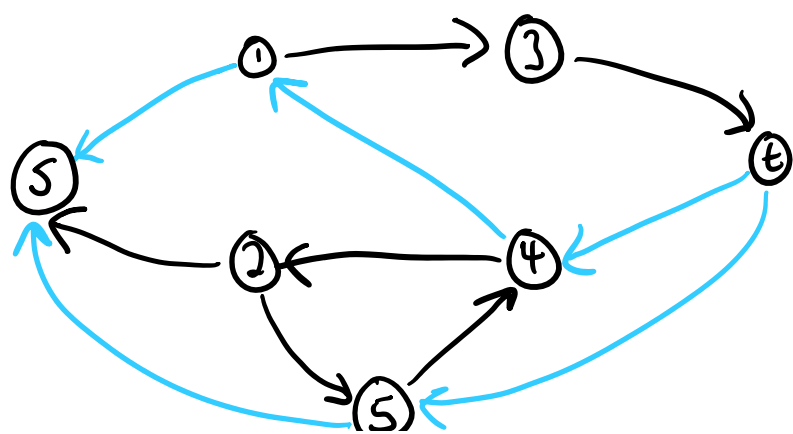
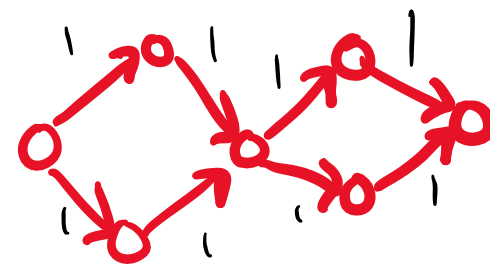
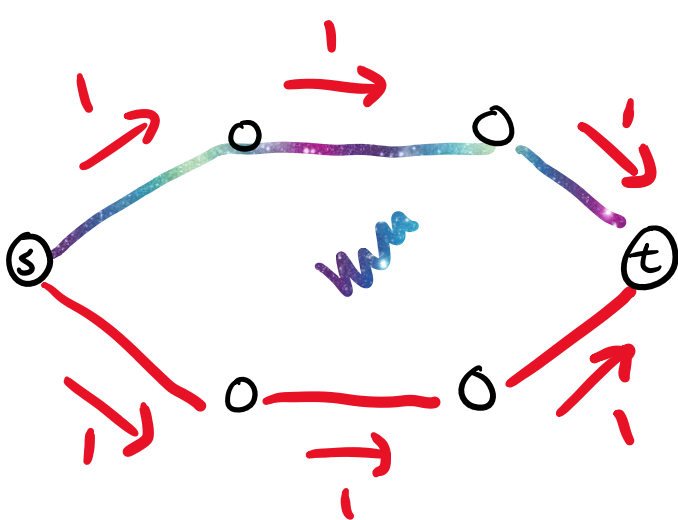
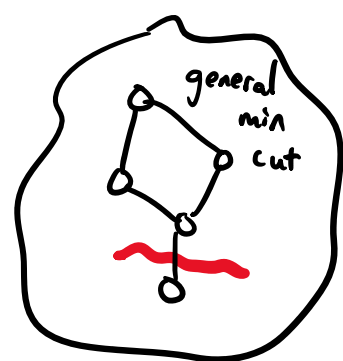


maximize

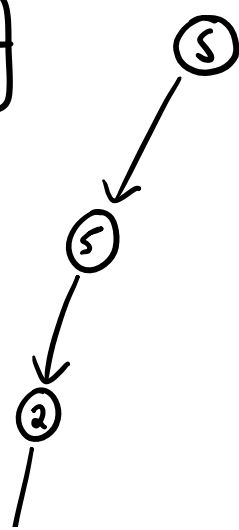
st-cut

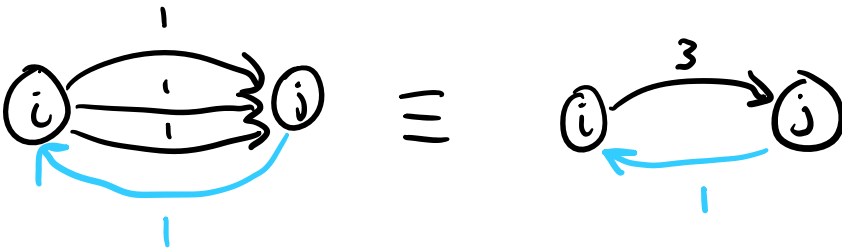


+2

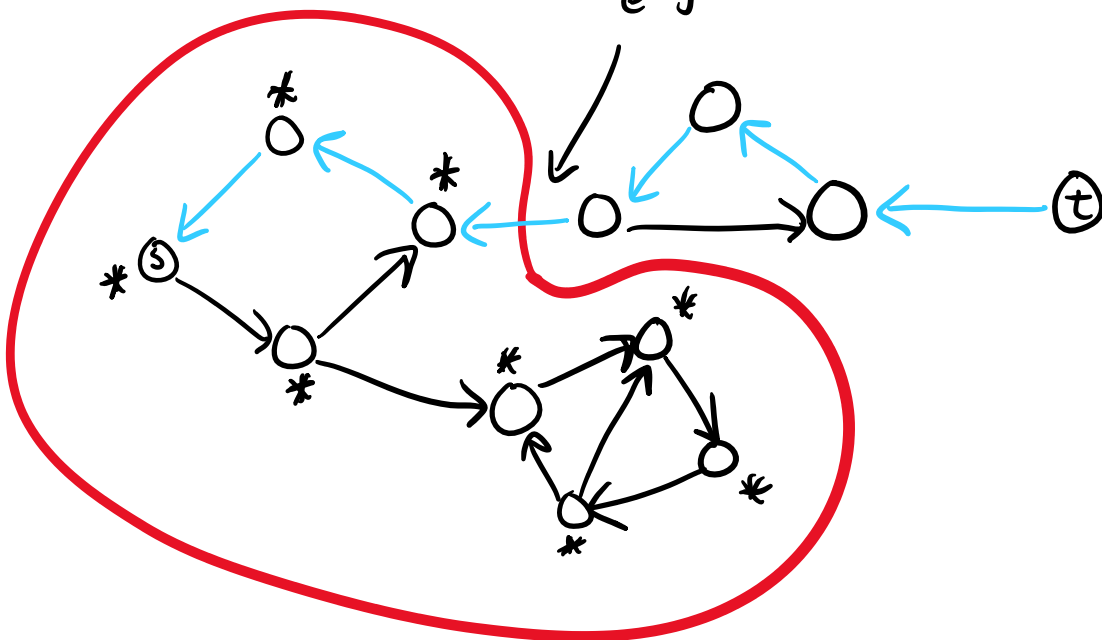


score
11





edges cross cut



Algorithm

while (canHazFlow())

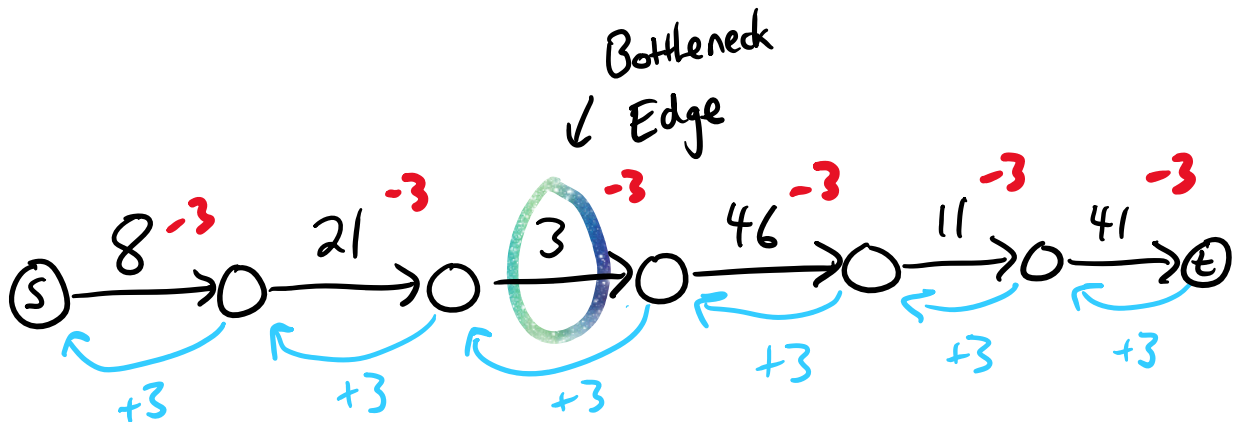
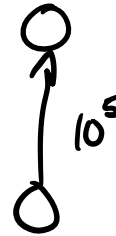
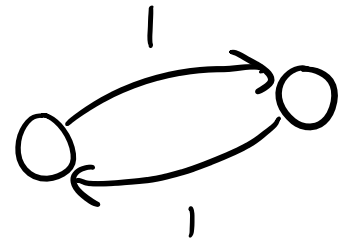
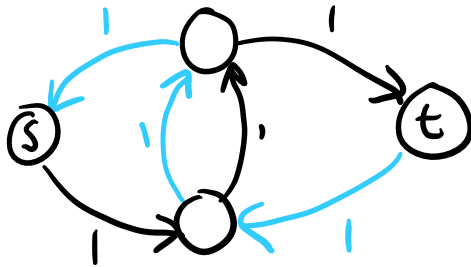
} Ford
Fulkerson

```

while (CanFindFlow())
{
    send Flow();
}

```

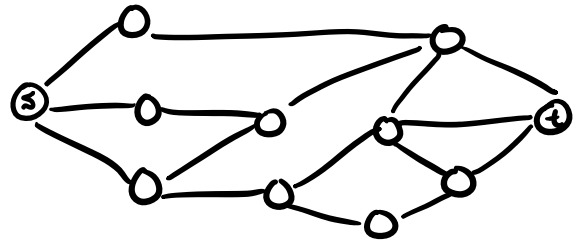
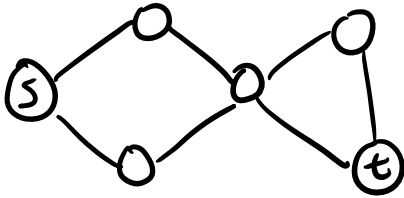
Fulkerson



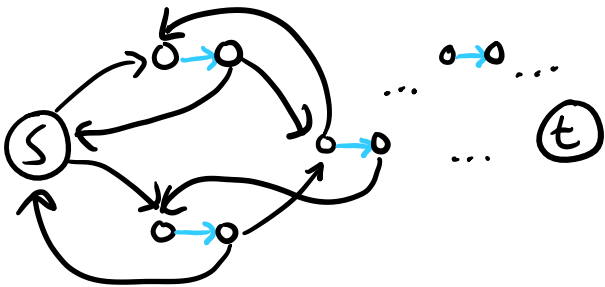
Runtime

$$O(f \cdot E)$$

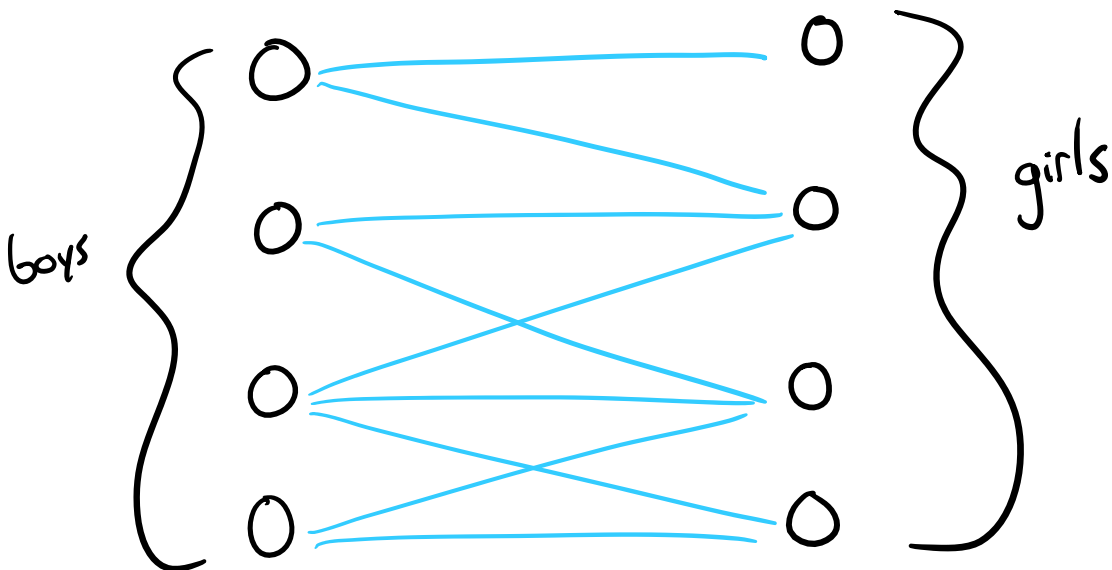
Vertex Disjoint Paths

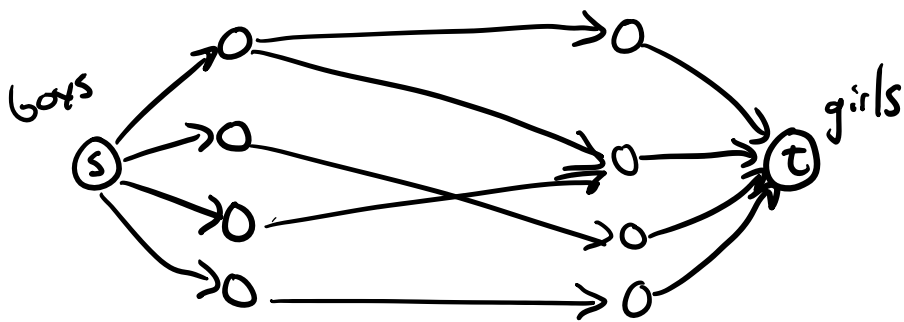


⇓ splitting the vertex

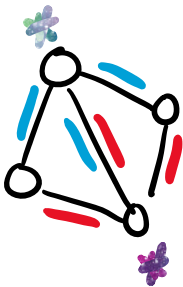


Bipartite Matching



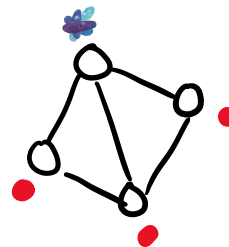
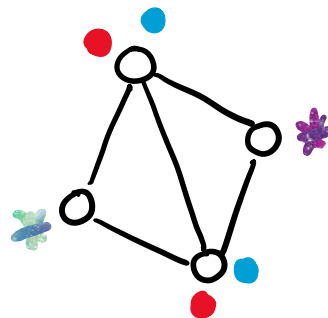


Vertex Cover

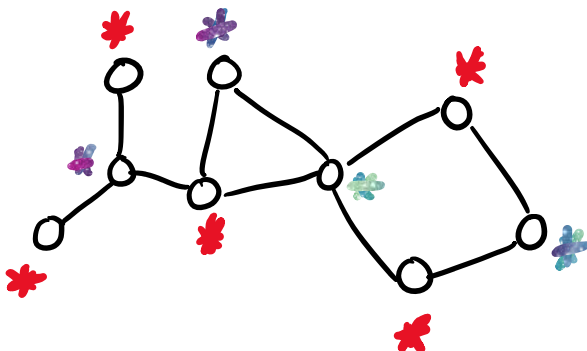


min #
of vertices

Dominating Set



Maximum Independent Set



Duality

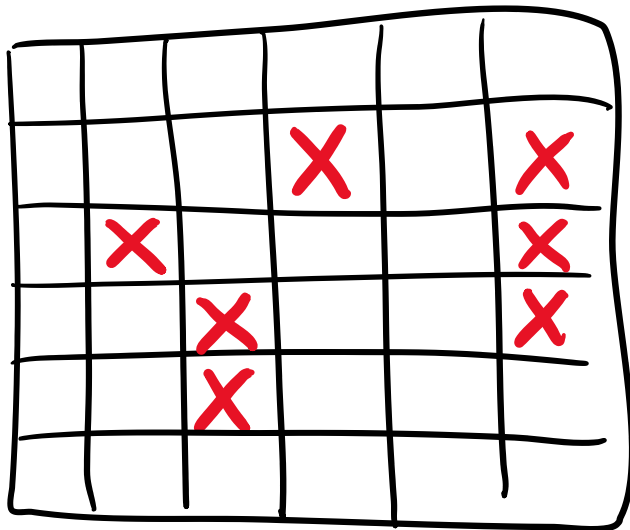
Max IS \leftrightarrow Vertex Cover

Königs Theorem

Bipartite Graph

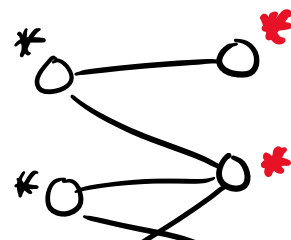
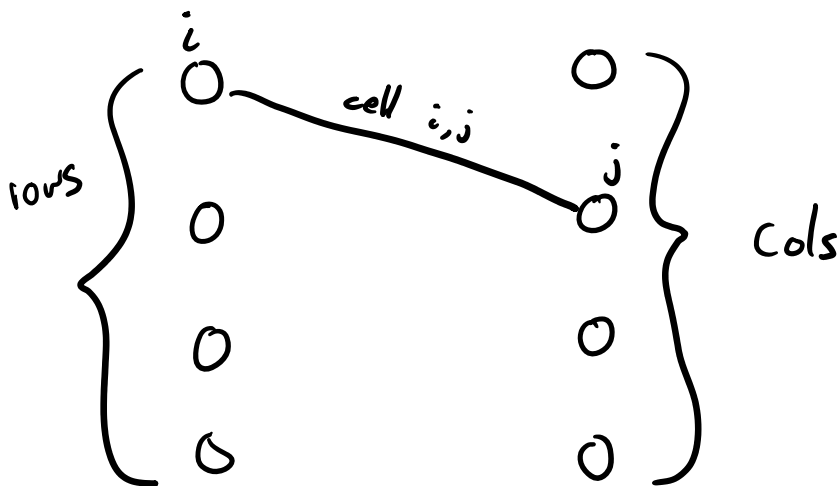
2-colorable

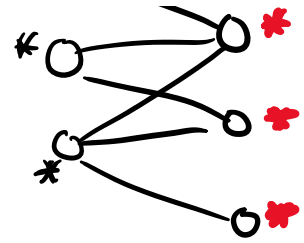
Min Vertex Cover = Maximum Bipartite Matching

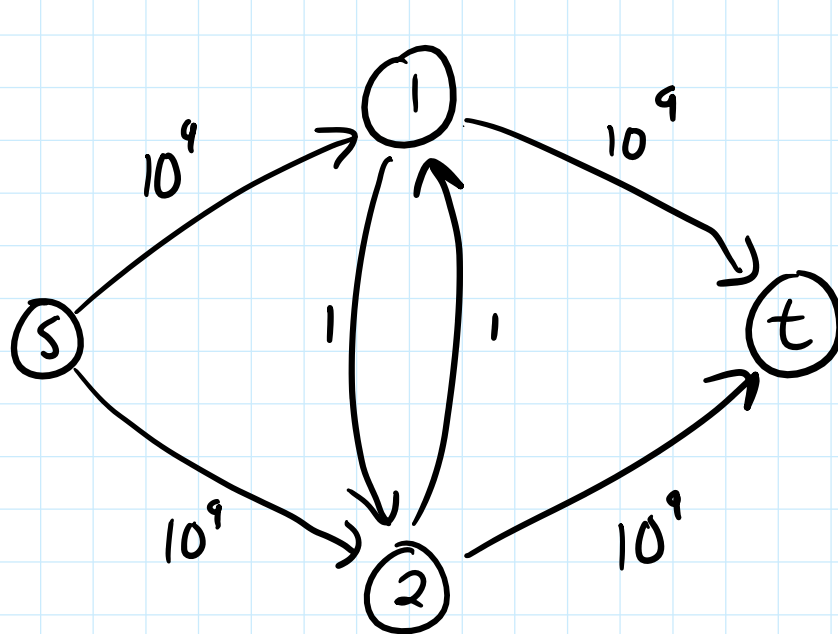


Max # Rooks
s.t. no two
attack

Cells \Rightarrow vertices







$O(f \cdot E)$
 ↙ BFS
 $O(E^2 V)$

B W W B
 W B W W
 W

