Some Remarks on flows

Theorem (Integrality theorem)

If $N = (V, E, c)$ is a network with an integer-valued capacity function $c: E \to \mathbb{Z}_+$ and $s \neq t$ are distinct vertices of V. Then there is an integer-valued maksimum flow in N (so $f: E \to \mathbb{Z}_0$).

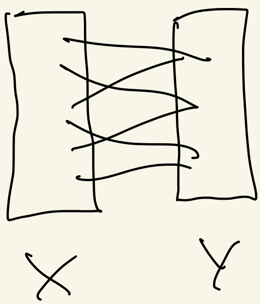
P: This follows from the way the Ford Folkerson algorithm works by induction over the # augmenting paths used before the algorithm has found a maximum flow:

- initially $f \equiv 0$ so clearly integer-valued
- assume the flow f_i after i augmenting paths is integer-valued
- In N_{f_i} all capacities are integers as c and f_i are integer functions. So $\delta(P_{i+1})$ is an integer where P_{i+1} is the $(i+1)$'st augmenting path
- This implies that f_{i+1} is integer-valued since we just add or subtract $\delta(P_{i+1})$ on some edges

$\square$.

Cormen 26.3 Bipartite matching

A graph $G=(V,E)$ is **bipartite** if we can partition its vertices in two sets X, Y s.t. all edges in E have an end in X and the other in Y



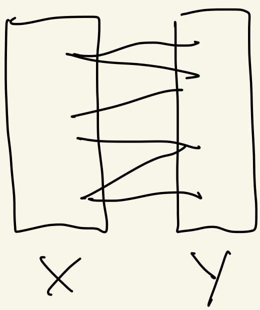
we also denote a bipartite graph $G=(X,Y,E)$

A **matching** is a set of edges, no two of which share an end-vertex:

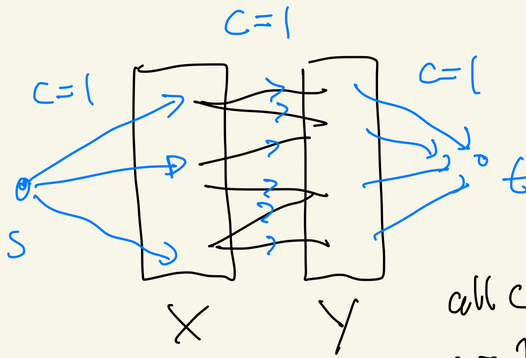
The **maximum matching** problem is given a graph $G=(V,E)$ Find a matching $M \subseteq E$ of maximum size

For non-bipartite graphs this is a difficult problem but it can be solved in polynomial time

Bipartite case reduction to a flow problem



$G=(X,Y,E)$



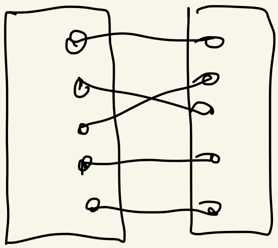
all capacities are 1

$N(G)$

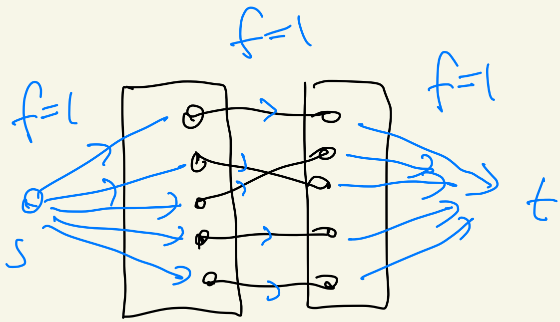
Theorem: The size of a maximum matching M^* in G is equal the the value of a maximum flow f^* in $N(G)$

proof

(a) $|f^*| \geq |M^*|$



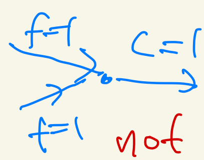
M^*



send 1 unit of flow along each of the $|M^*|$ blue (disjoint) paths

(b) $|f^*| \leq |M^*|$

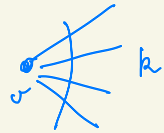
- By the integrality theorem there is an integer valued maximum flow f^* in $N(G)$
- This will send one unit of flow on exactly $|f^*|$ edges from X to Y and these form a matching!



not possible

A neat application of the integrality theorem

<u>Definition</u> a graph $G=(V,E)$ is **k-regular** if $d(v)=k$ $\forall v\in V$, that is, each vertex is incident to k edges



<u>Theorem (König 1930's)</u>

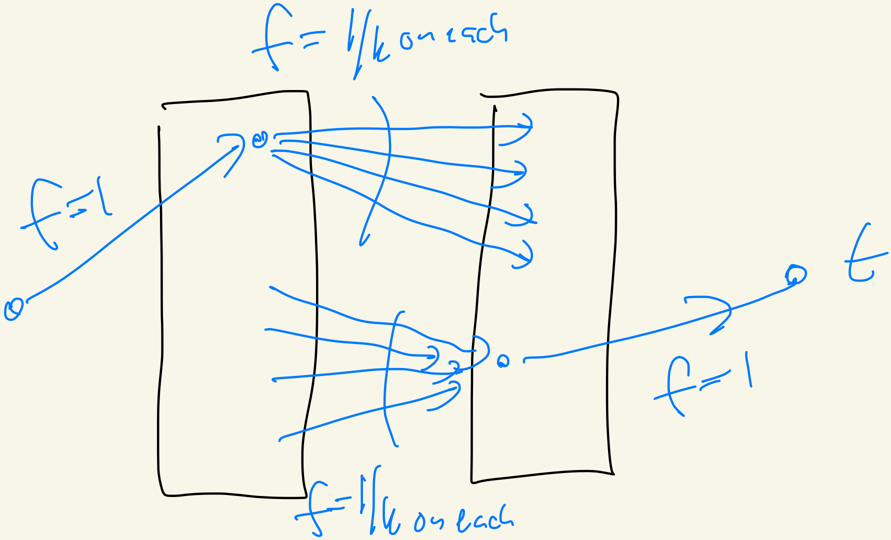
For every $k\geq 1$ every k-regular bipartite graph $G=(X,Y,E)$ has a perfect matching ($|M|=|X|=|Y|$)

<u>P:</u> First observe that if G is k-regular, then $|X|=|Y|$: count E in two ways

$$|X|\cdot k=|E|=|Y|\cdot k \Rightarrow |X|=|Y|$$

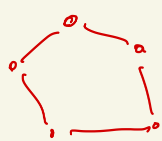
Consider $N(G)$ and let f be the flow that has $f(s,x)=1=f(y,t)$ $\forall x\in X, y\in Y$

$$f(xy)=\frac{1}{k} \quad \forall xy\in E$$



- Then $|f| = |X| = c(s, V\setminus\{s\})$ so f is a maximum flow
- By the integrality theorem, there exists an integer-valued maxflow f^* with $|f^*| = |X|$

So by the previous theorem G has a matching M of size $|X|$ $\square$.



2-regular
no perfect matching

Flows with general balances

Suppose we are given a network $N=(V,E,c)$ and an additional integer-valued function $b: V \to \mathbb{Z}$

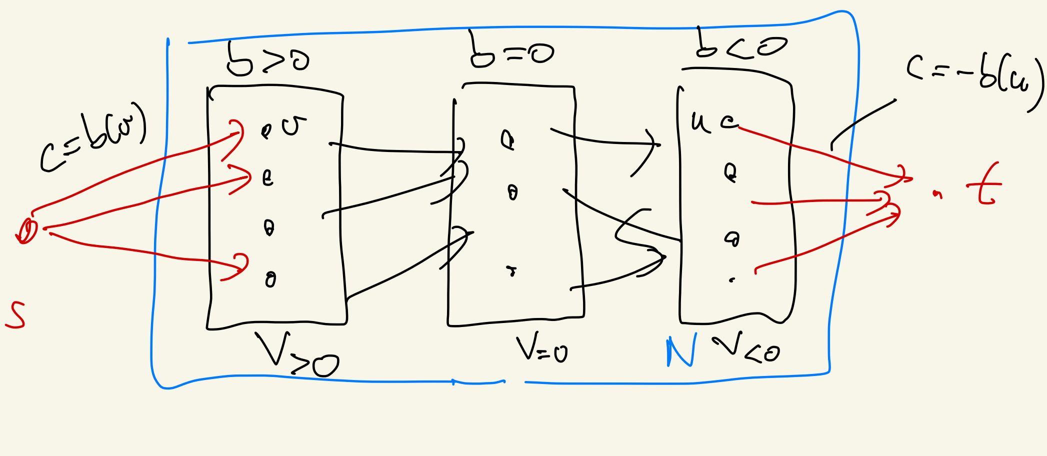
Denote N by $N=(V,E,c,b)$

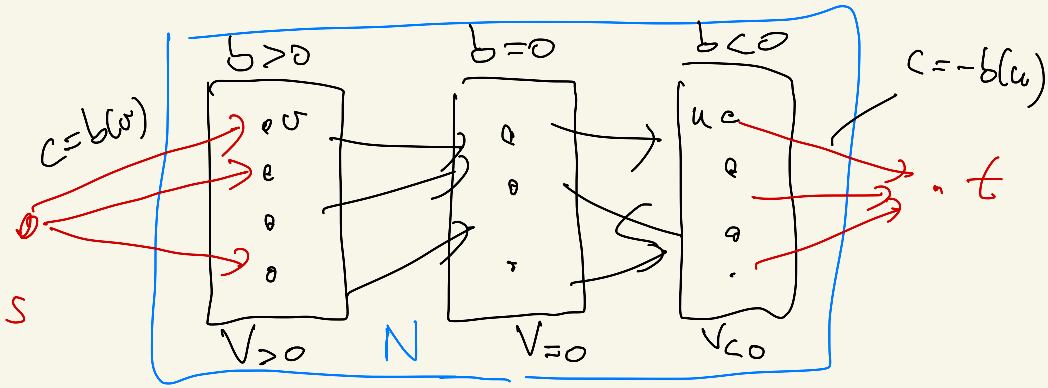
Question: Does there exist a flow f in N with
$0 \le f(u,v) \le c(u,v)$ $\forall (u,v) \in E$ AND $b_f(v) = b(v)$?

As we know $\sum_{v \in V} b_f(v) = 0$ for every flow

we can assume that $\sum_{v \in V} b(v) = 0$

Construct a network N' from N as follows





<u>Claim</u> N has a flow f with
$0 \le f \le c$ and $b_f \equiv b$
if and only if N' has an (s,t)-flow
of value $\sum_{\{v \mid b(v) > 0\}} b(v)$

<u>P:</u> $\Rightarrow$: if f satisfies $b_f \equiv b$ and $0 \le f \le c$ in N
then f' which equals f on edges in N
and has $f'(s,v) = b(v) \ \forall v \in V_>$
$f'(u,t) = -b(u) \ \forall u \in V_<$
is an (s,t)-flow in N' of value $\sum_{\{v \mid b(v) > 0\}} b(v)$

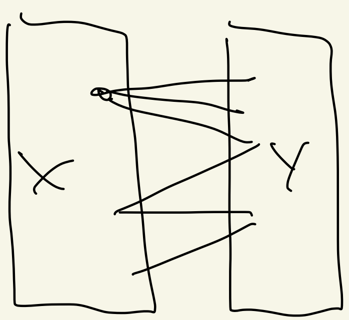
$\Leftarrow$: if f' is an (s,t)-flow of value

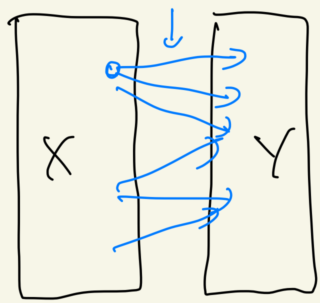
$$\sum_{\{v \mid b(v) > 0\}} b(v) \quad \text{in } N'$$

then the flow $f'|_E$ (f' restricted to E)

is a flow in N with $b_f(v) = b(v)$ $\square$.

<u>Remark</u>: The problem of deciding whether a bipartite graph $G = (X, Y, E)$ with $|X| = |Y|$ has a matching of size $|X|$ (called a perfect matching) is the same as asking for a flow f with $b_f(v) = 1$ if $v \in X$ and $b_f(v) = -1$ if $v \in Y$ in the following network:



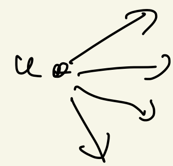


$\tilde{N}(G)$

Orienting a graph to a directed graph with specified out-degrees

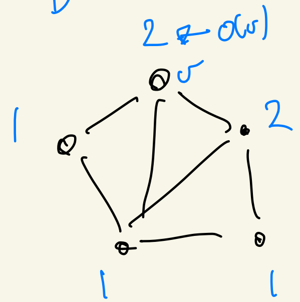
Definition An orientation of a graph $G=(V,E)$ is any directed graph $D=(V,A)$ that we can obtain from G by assigning each edge $(u,v)\in E$ one of the possible orientations $u\to v$ or $v\to u$

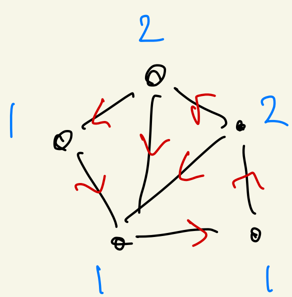
The out-degree, $d^+_D(u)$ of a vertex in a digraph is the number of arcs going out of u



Orientation problem: Given $G=(V,E)$ and a function $o: V\to \mathbb{Z}_0$ s.t $\sum_{v\in V} o(v) = |E|$

Does there exist an orientation D of G such that $d^+_D(v) = o(v)$ for all $v\in V$?





Formulation as a flow problem

Given $G=(V,E)$ and $o: V \to \mathbb{Z}_0$ with $\sum_{v \in V} o(v) = |E|$

make a reference orientation D' of G by arbitrarily orienting each edge of G. Let A' be the arcs of D' and

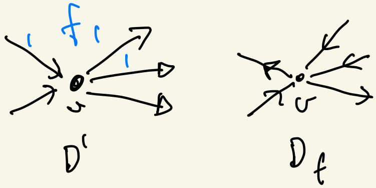
Interpret an integer (0-1) flow on the arcs A' as follows

$f(uv)=1 \Rightarrow$ reverse uv

$f(uv)=0 \Rightarrow$ keep orientation uv

Call the resulting orientation of G D_f. Then

$$d^+_{D_f}(v) = d^+_{D'}(v) - \sum_{(v,w)\in A'} f(v,w) + \sum_{(u,v)\in A'} f(u,v)$$



$$= d^+_{D'}(v) - b_f(v)$$

Hence D_f is a good orientation of G precisely when $o(v) = d^+_{D_f}(v) = d^+_{D'}(v) - b_f(v)$

$$\Updownarrow$$

$$b_f(v) = d^+_{D'}(v) - o(v)$$

So G has the desired orientation if and only if the network $N' = (V, A', c \equiv 1, b)$ has a flow f with $0 \le f \le 1$ and $b_f(v) = b(v)$, where $b(v) = d^+_{D'}(v) - o(v)$

Remarks

- D' was our fixed (arbitrary) orientation so $d^+_{D'}(v)$ is a constant for each v
- $b(v) = d^+_{D'}(v) - o(v)$ is also a constant for each v

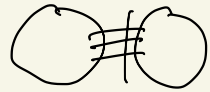
and
$$\sum_{v\in V} b(v) = \sum_{v\in V}\left(d^+_{D'}(v) - o(v)\right)$$
$$= \sum_{v\in V} d^+_{D'}(v) - \sum_{v\in V} o(v)$$
$$= |E| - |E| = 0$$

so b is a valid balance function

Note that we can find the desired orientation when it exists by finding an integer valued feasible flow f and reversing those arcs of D' for which $f(u,v)=1$

Determining the edge-connectivity using flows

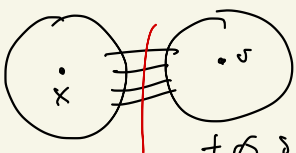
- Recall that the edge-connectivity, $\lambda(G)$, of a graph G is the minimum # edges whose removal disconnects G (so size of a minimum cut)



min # edges

- Let $\lambda(x,y)$ be the minimum size of an (x,y)-cut, that is min # edges whose removal kills all (x,y)-paths
- Then $\lambda(G) = \min\{\lambda(x,y) \mid x,y \in V\}$.

In fact for every fixed vertex x we have $\lambda(G) = \min\{\lambda(x,v) \mid v \in V - x\}$



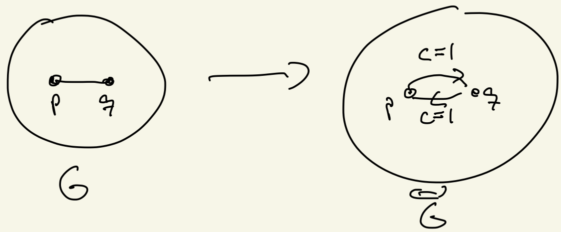
$\neq \emptyset$ so some v here

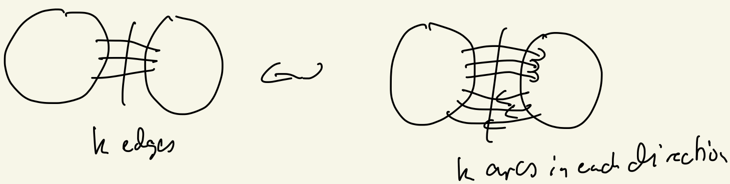
$\lambda(G)$ edges

- Conclusion: it is enough to find the values $\{\lambda(x,v) \mid v \in V\}$ and take the minimum

How to determine $\lambda(x,v)$?

Use flows! :)





So we are looking for a min capacity (x,v)-cut in $\overleftrightarrow{N} = (V, \overleftrightarrow{E}, c \equiv 1)$

rename x to s and v to t

Now we look for a minimum (s,t)-cut in a network

By the max flow min cut thm, the capacity of a minimum (s,t)-cut in $\overleftrightarrow{N}$ equals the value $|f^*|$ of a maximum (s,t)-flow

So for each $v \in V - x$ we can determine $\lambda(x,v)$

There are $|V|-1$ choices for $v \in V \setminus \{x\}$

so we need $|V|-1$ max flow calculations

to determine $\{\lambda(x,v) \mid v \in V \setminus \{x\}\}$

and then we get $\lambda(G)$ as

$$\lambda(G) = \min\{\lambda(x,v) \mid v \in V \setminus \{x\}\}$$

Remark: In Cormen  are not allowed but we can easily change the digraph

Increases running time but still polynomial

(and we don't need to do it as we can redefine the residual network to work when there are .)