CS7800: Advanced Algorithms

Lecture 8: Applications of Network Flow

- Bipartite matching
- Image segmentation

Jonathan Ullman

10-04-22

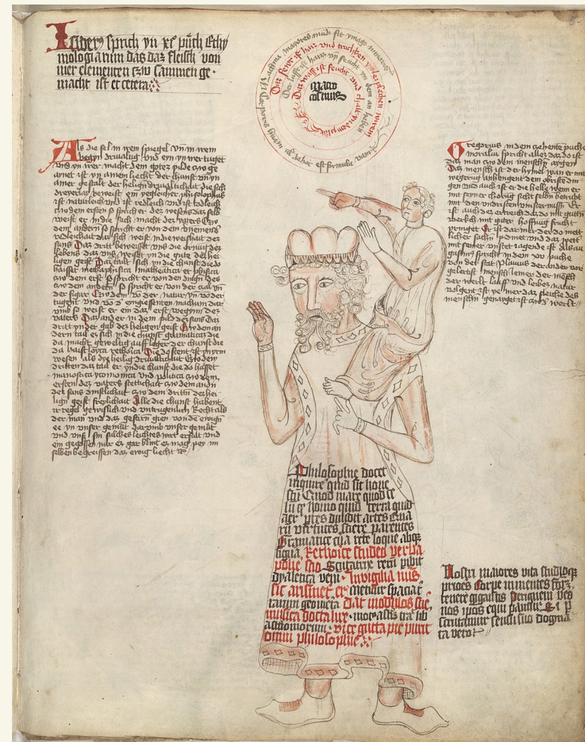
How many more algorithms do we really need?

Last week:

- Can find a max flow or a min cut in $O(mn)$ time

Actually can do better!

- If G has integer capacity we can find an integer max flow



Today: Standing on the shoulders of giants

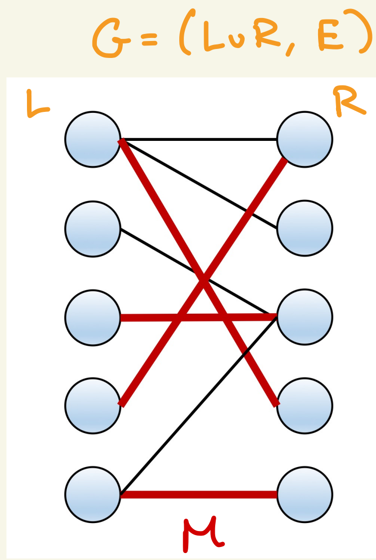
Maximum Cardinality Bipartite Matching

two types of nodes

Input: A bipartite graph $G = (L \cup R, E)$

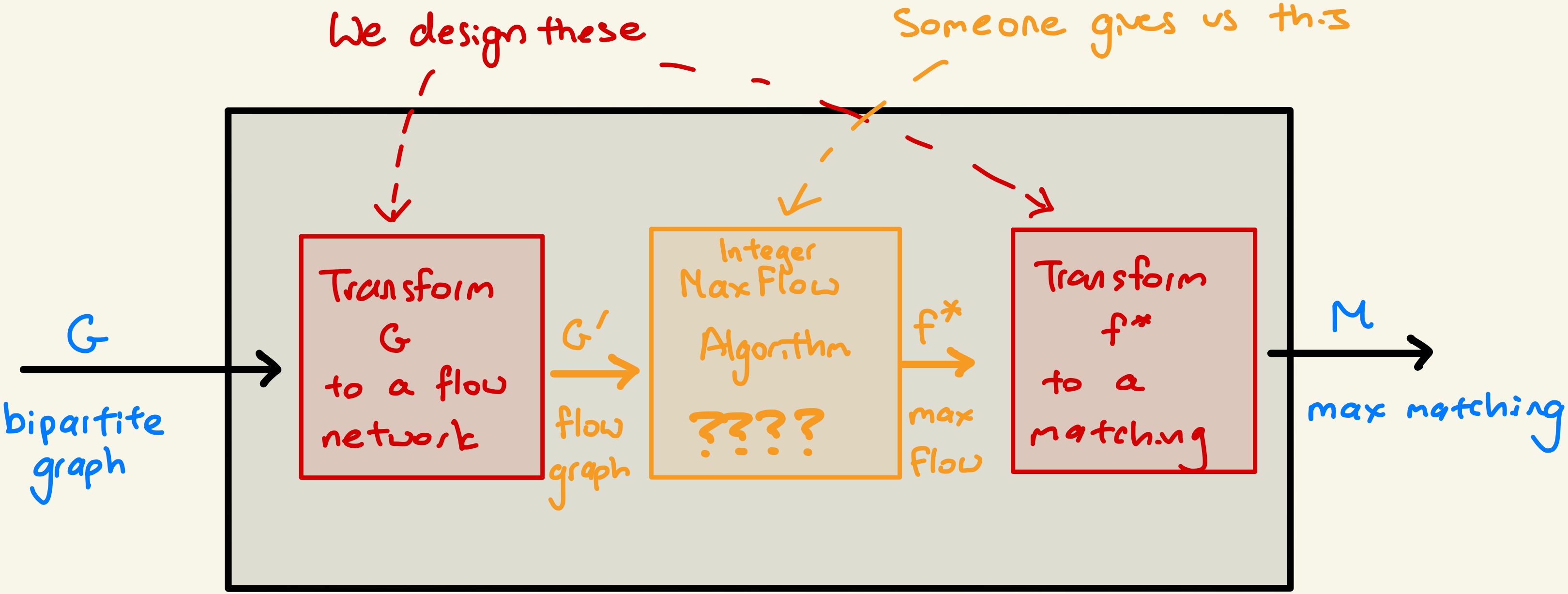
Output: A matching $M \subseteq E$ of maximum size

no edges in E share an endpoint



- A matching M is perfect if $|M| = |L| = |R|$

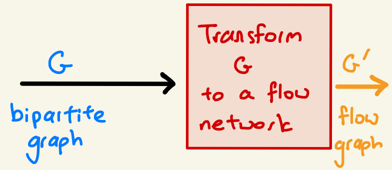
Reductions

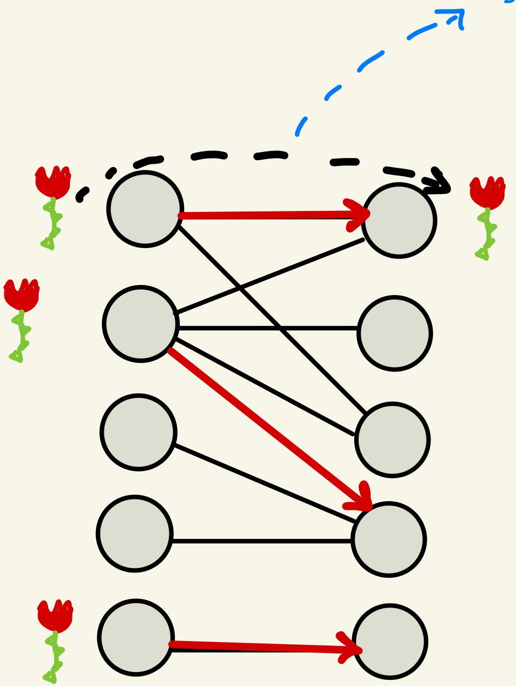


Nothing special about matching and max flow

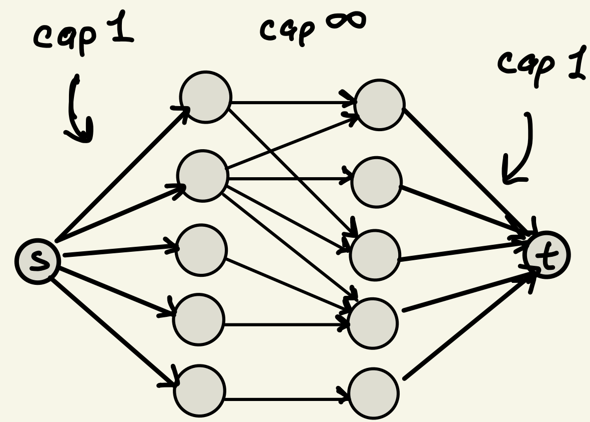
There are other more general kinds of reduction

Reducing Matchings to Flows



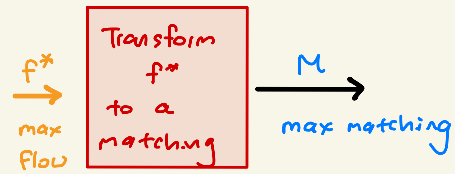


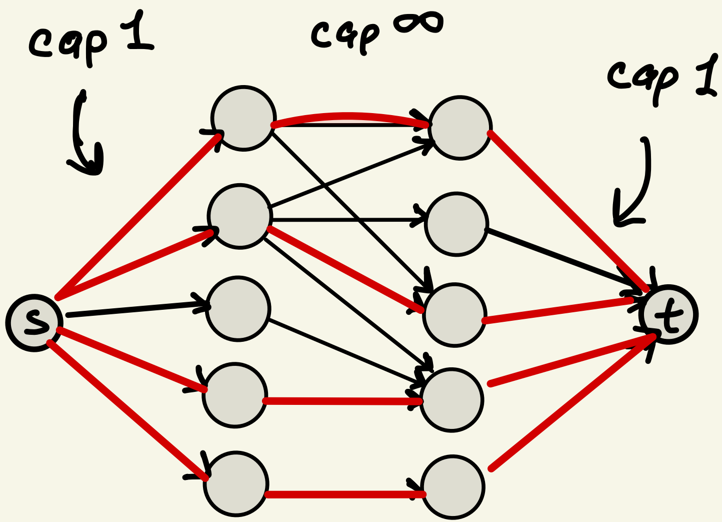
- We produce a valid integer flow network
- Can do the transformation in $O(m)$ time



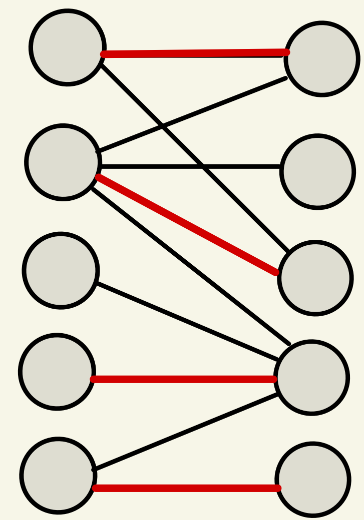
Reducing Matchings to Flows

- We obtain a matching
- Can transform in $O(m)$ time

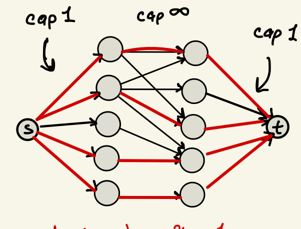




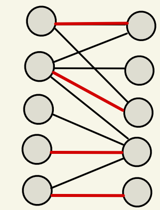
red edges have flow 1



Correctness of the Reduction



red edges have flow 1



① M is a matching ✓

② If there is a matching of size k then $val(f^*) \geq k$

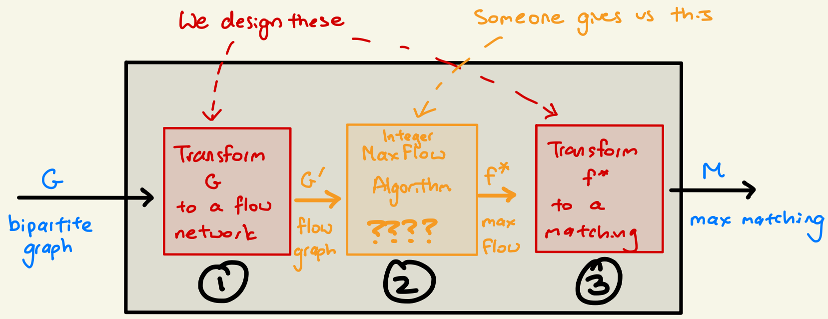
✓

③ If $val(f^*) = k$ then there is a matching of size $\geq k$

✓

size of M $= val(f^*)$

Running Time Analysis



#of nodes/edges in G'

Assume that we solve max flow in time $O(m'n')$

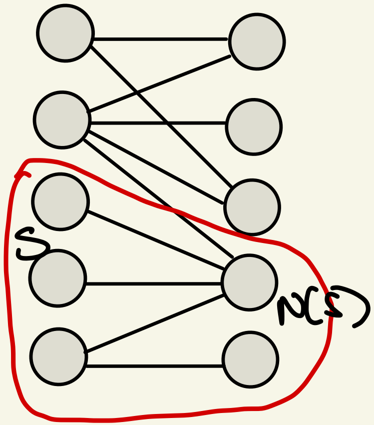
① takes $O(m)$ time, outputs G' with $n' = 2n+2$ $m' = m+2n$

② takes $O(m'n') = O((m+n)(n+1)) = O(mn)$

③ takes $O(n)$ time

Algorithm takes $O(mn)$ time

Hall's Theorem



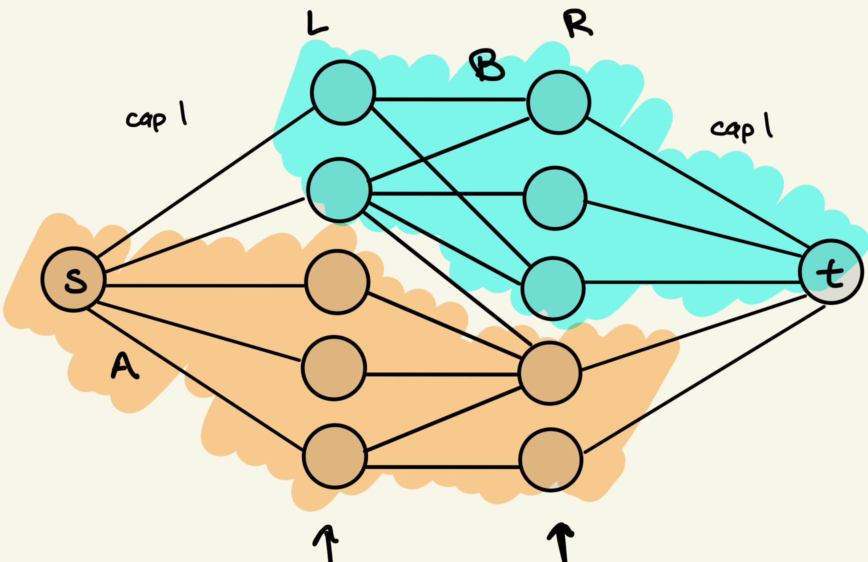
Why cant this graph have a perfect matching?

For $S \subseteq L$ (or $S \subseteq R$) let $N(S)$ = set of nodes on the R (or L) connected to S

<u>Thm:</u> G has a perfect if and only if $|N(S)| \geq |S|$ for every $S \subseteq L$ or $S \subseteq R$

We will prove: if G <u>doesn't</u> have a perfect matching then $\exists$ S such that $|N(S)| < |S|$.

Proof of Hall's Theorem



$S = L \cap A$ $\qquad$ $N(S) \subseteq A$

Observations:

① Min cut doesn't include edges with cap ∞

② No perfect matching

$\Downarrow$

$$\boxed{\begin{array}{c}\text{Max flow} < n \\ \Downarrow \\ \text{Min cut} < n\end{array}}$$

Use the duality thm

$$|S| + |L \setminus S| = n$$
$$|N(S)| + |L \setminus S| = cap(A, B) < n \implies |N(S)| < |S| \quad \square$$

Image Segmentation

<u>Goal</u>: Separate image into foreground and background

How?

Image Segmentation

Input: An undirected graph $G = (V, E)$

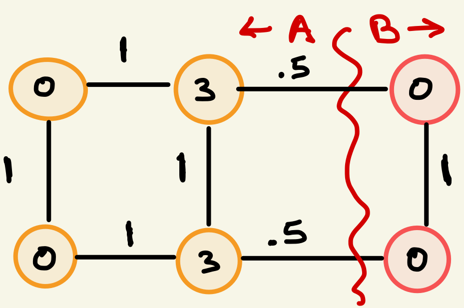
Likelihoods $a_v, b_v \geq 0$ for each node

Separation penalty $p_{uv} \geq 0$ for each edge

Output: A partition of V into (A, B) maximizing

$$\boxed{\text{quality}(A,B) = \sum_{v \in A} a_v + \sum_{v \in B} b_v - \sum_{\substack{(u,v) \in E \\ u \in A \;\; v \in B}} p_{uv}}$$

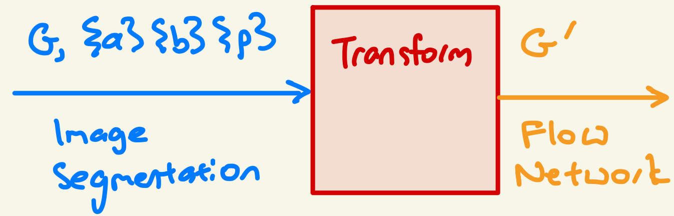
Example: (All $b_v = 0$)



$\text{quality}(A,B) = 6 + 0 - 1$

Designing the Reduction to Min Cut

$$\boxed{\text{quality}(A,B) = \sum_{v \in A} a_v + \sum_{v \in B} b_v - \sum_{\substack{(u,v)\in E \\ u\in A \;\; v \in B}} p_{uv}}$$



$$\max \; \text{quality}(A,B)$$

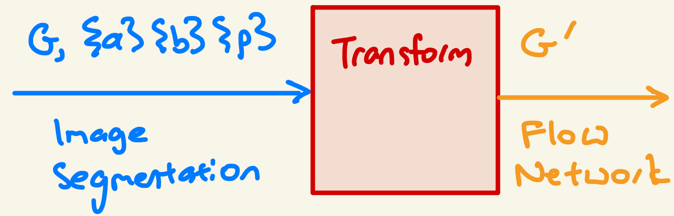
$$\Downarrow$$

$$\min \; -\text{quality}(A,B) = \left(\sum_{v\in A} -a_v\right) + \left(\sum_{v \in B} -b_v\right) + \sum_{(u,v)\in E} p_{u,v}$$

$$\Downarrow$$

$$\boxed{\min \; -\text{quality}(A,B) = \sum_{v\in B} a_v + \sum_{v\in A} b_v + \sum_{\substack{(u,v)\in E \\ u\in A \;\; v\in B}} p_{u,v}}$$

Designing the Reduction to Min Cut

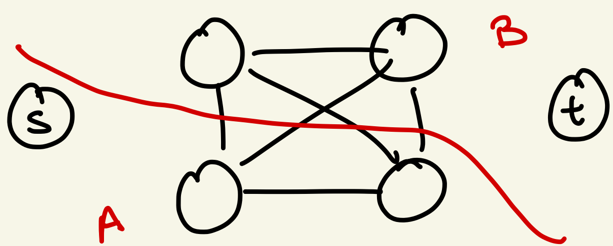


min cut problem is

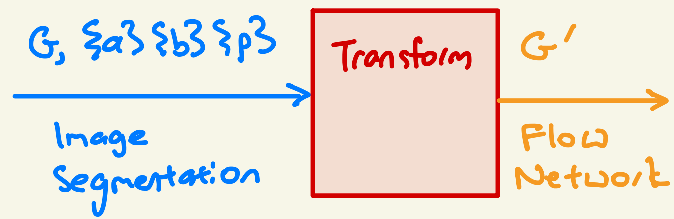
$\min_{A,B}$ s.t. $s \in A$ $t \in B$

$$\min_{A,B} -\text{quality}(A,B) = \sum_{v \in B} a_v + \sum_{v \in A} b_v + \sum_{\substack{(u,v) \in E \\ u \in A \; v \in B}} p_{u,v}$$

① Min cut requires $s \in A$ $t \in B$ $\Longrightarrow$ Add "dummy nodes" s, t

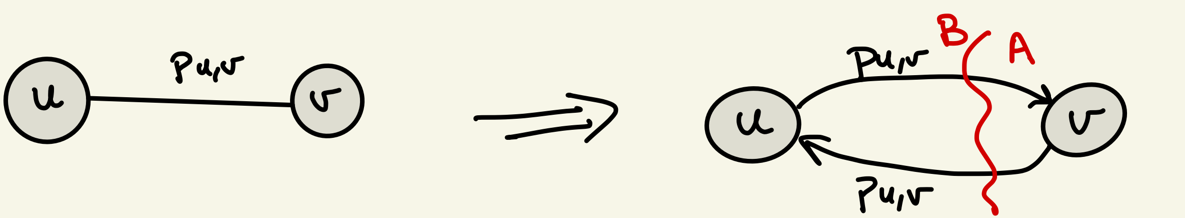


Designing the Reduction to Min Cut

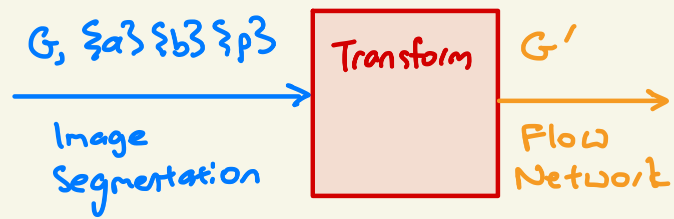


$$\min_{A,B} \; -\text{quality}(A,B) = \sum_{v \in B} a_v + \sum_{v \in A} b_v + \sum_{\substack{(u,v) \in E \\ u \in A \; v \in B}} p_{u,v}$$

② Min cut treats A,B asymmetrically $\left(\text{cap}(A,B) = \sum_{\substack{u \to v \\ u \in A \; v \in B}} \text{cap}(u \to v) \right)$

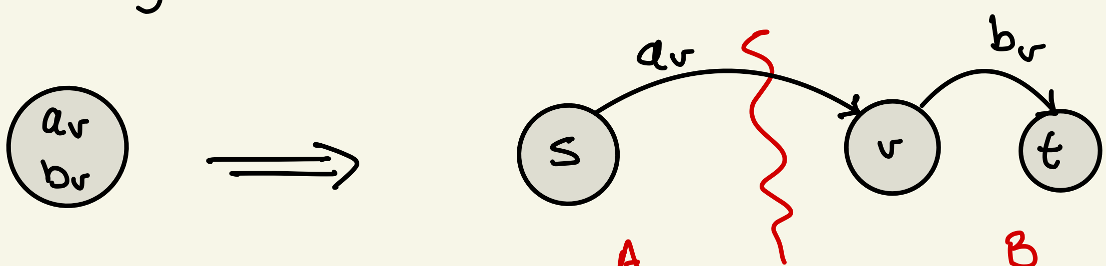


Designing the Reduction to Min Cut

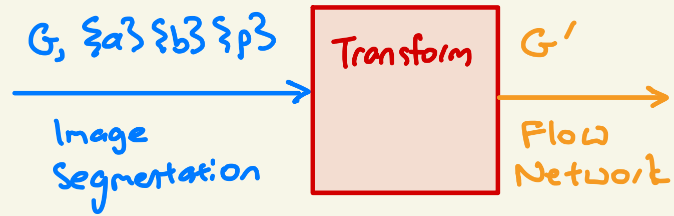


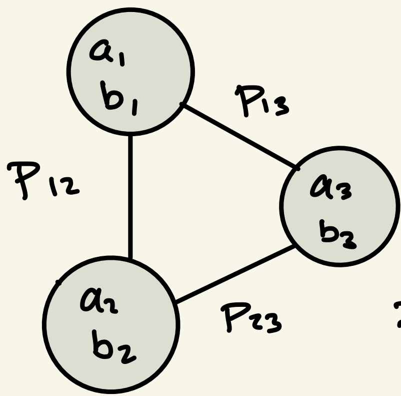
$$\min_{A,B} \; -\text{quality}(A,B) = \sum_{v \in B} a_v + \sum_{v \in A} b_v + \sum_{\substack{(u,v) \in E \\ u \in A \; v \in B}} p_{u,v}$$

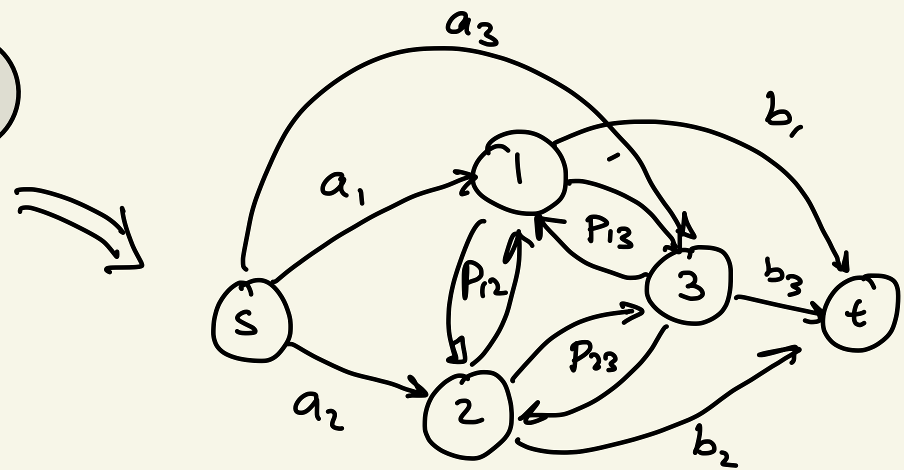
③ Min cut only has edge capacities



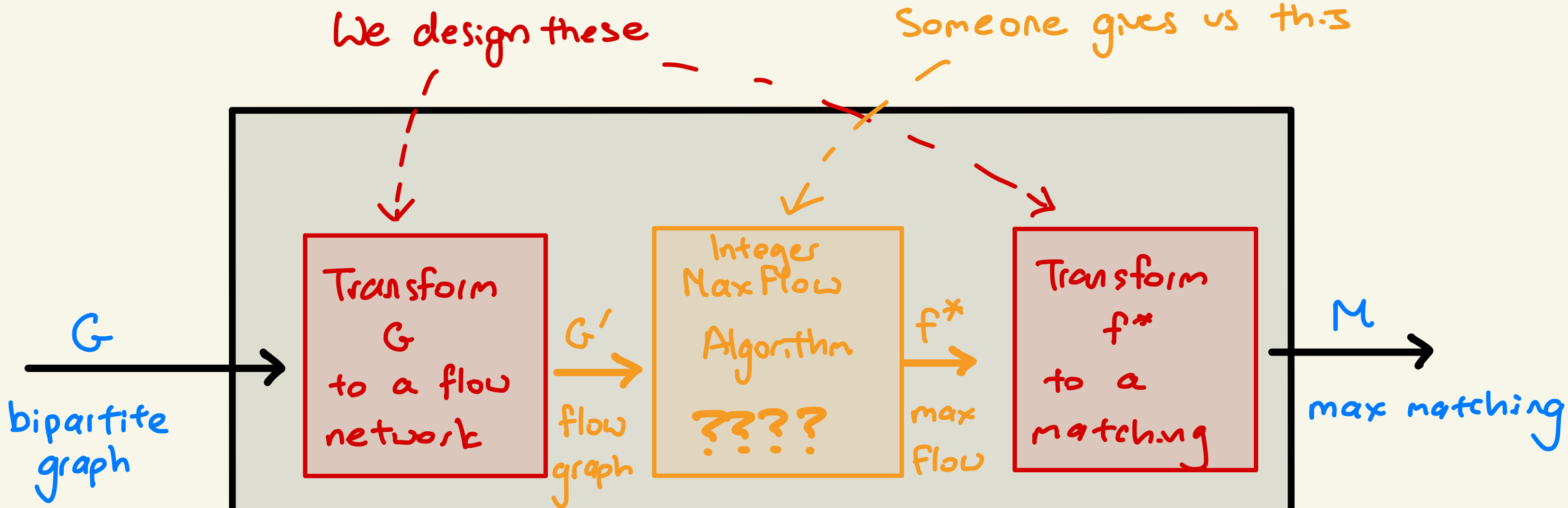
Designing the Reduction to Min Cut







Reductions



Nothing special about matching and max flow

There are other more general kinds of reduction