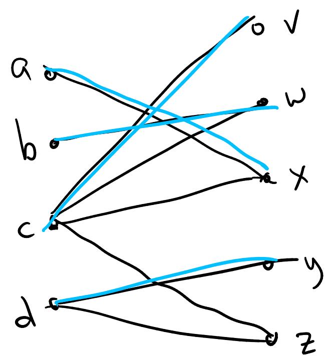
Max-Flow Applications.

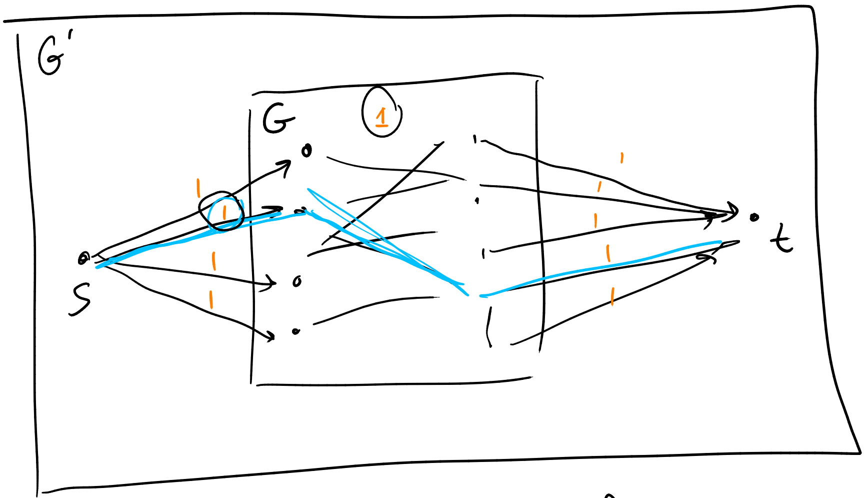
① Max bipartite matching.



Matching = set of edges that do not share any endpoints.

Max matching = greatest possible # of edges in a matching.

Claim: we can solve this with max flow.



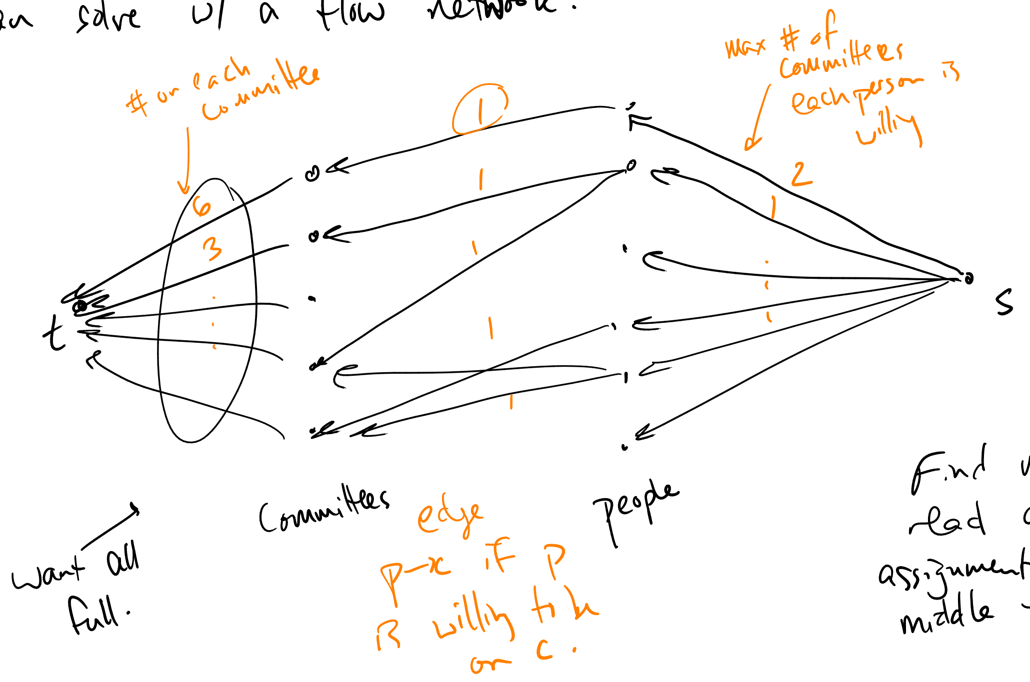
Claim: any valid flow of value k from s → t corresponds to a matching of size k on G, and vice versa.

Edges w/ flow on them are edges in the matching.
Restrictions on capacity + flow rules ensure that we can't have multiple flows going out of or into same vertex.

Hence a max flow + max matching are the same.

② Committee assignments. Have:
- A set of committees.
- A set of people.
- Each committee needs a certain # of people.
- Each person has a set of committees they are willing to be on and a max # of committees they are willing to be on.
- Goal: assign people to committees in a way that respects everyone's preferences, and fills the committees.

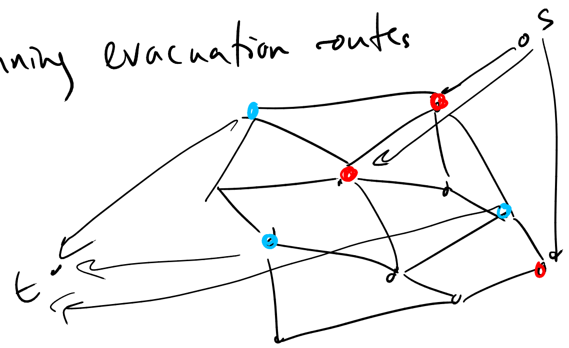
Can solve w/ a flow network!



Find max flow, read off committee assignments from edges in middle that have flow.

Other examples:

- Schedule final exams: match classes — rooms — times
- Planning evacuation routes



- Baseball Elimination
- Image segmentation
- Many others.

GreedyFlow(G, s, t):

While there is any s → t path in G with nonzero capacity:

P ← find an s → t path that does not use full edges.

α ← min capacity along P

for each edge e ∈ P:

f(e) ← f(e) + α