Network flow

Network flow

- Directed graph with distinguished vertices s (the source) and t (the sink)
- Edge (u,v) has capacity c(u,v)

Network flow

- Directed graph with distinguished vertices s (the source) and t (the sink)
- Edge (u,v) has capacity c(u,v)

- Flow f is function E -> numbers obeying:
  - Capacity: 0 <= f(u,v) <= c(u,v) for every edge (u,v)
  - Conservation: sum_u f(u,v) = sum_u f(v,u) for v != s,t

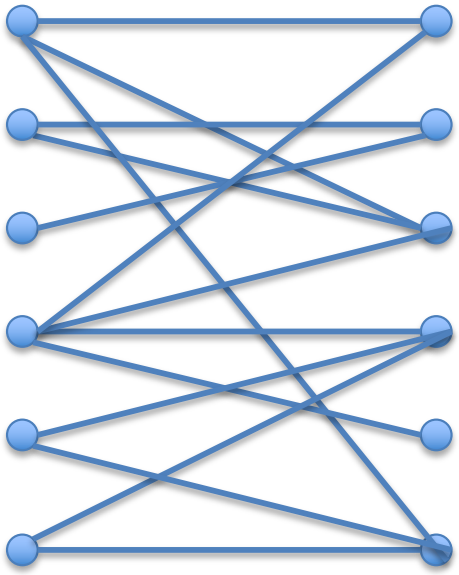
For now…

- Don’t worry about finding a maximum flow, but assume that we can do so efficiently

- In particular, if all capacities are integers, the flow on every edge will be an integer

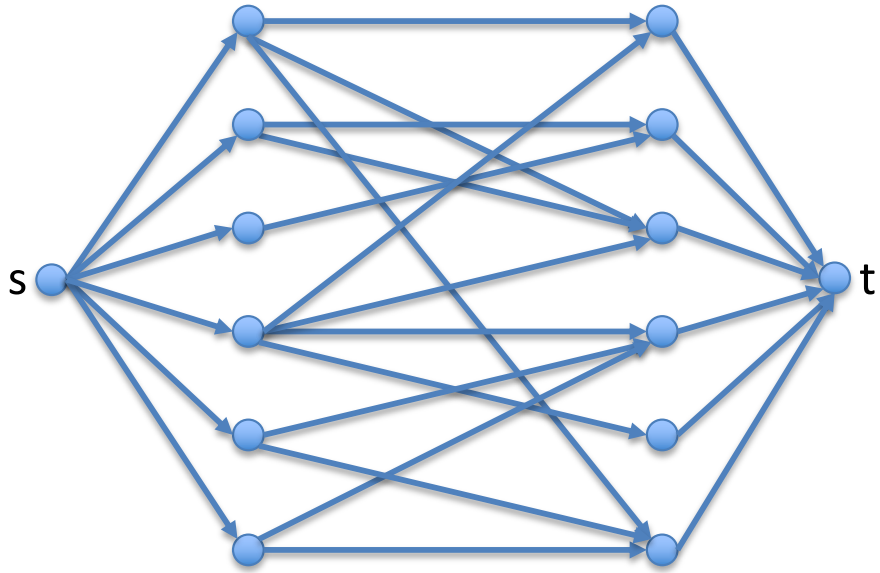
Bipartite matching

- Given bipartite graph, find largest set of edges so that none share an endpoint



Bipartite matching

- Given bipartite graph, find largest set of edges so that none share an endpoint



All edges have capacity 1

After graduation, your first job is to schedule commercials for a TV station. Specifically, the TV station has a collection of n shows. Show i has a number s_i advertising breaks, each of which can accommodate up to 3 commercials. The station also has a list of commercials, each of which comes with a subset of shows during which it can be advertised. (Some advertisers avoid shows with controversial or otherwise undesirable content.) Each commercial comes with a limit on the number of times it can be shown. In addition, no commercial can ever be shown more than once during a single advertising break.

Although the TV station must obey all these constraints, their goal is to maximize the number of commercials shown (counting each showing of a commercial separately) in order to maximize their advertising revenue. Give an algorithm based on network flow that finds a schedule for commercials that does this. Include a justification for why your algorithm works.