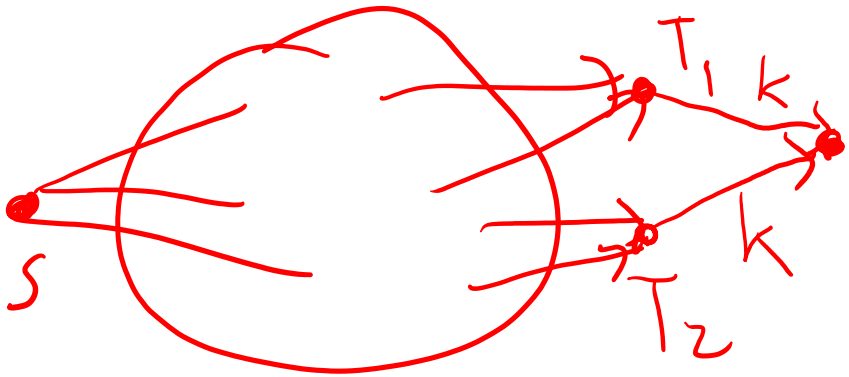
two targets T_1, T_2
want to max. flow to T_1, T_2 combined
(T_1 & T_2 have max capacities - can only process k gallons/second

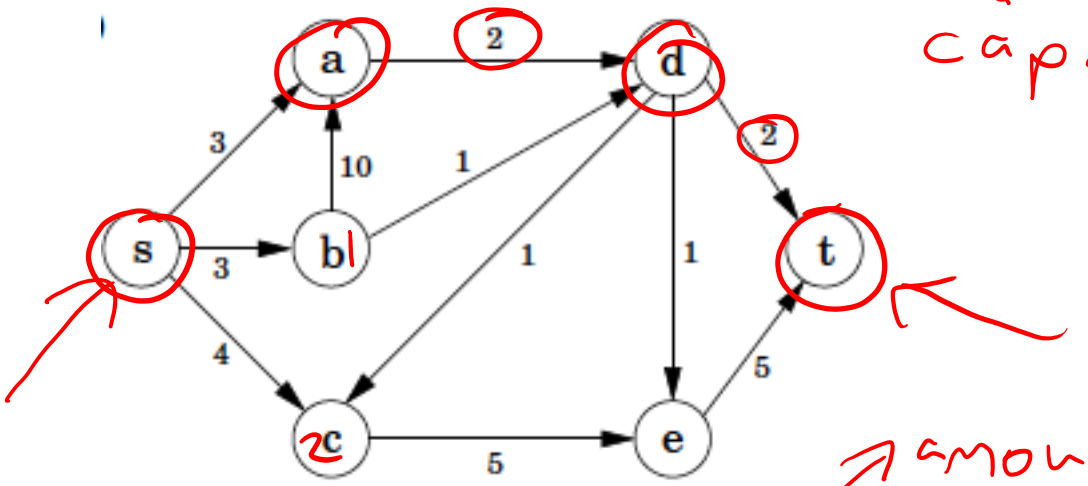
Network Flows



flow in = flow out

Network Flows

- You are given a network with capacities on the edges



cap / time

amount of resource

- Goal is to decide how much flow to send along each edge, to maximize total amount from s to t

Useful Strategy: Modifying the Input Graph

- Many problems with network flow require you to modify the input graph
- Example: Suppose you are given a graph with positive integer capacities. You are given a source node S, and two target nodes T_1, T_2. You want to maximize the total amount of flow sent to T_1 and T_2 combined.

- How can you use the existing flow algorithm on a modified graph to answer this question?

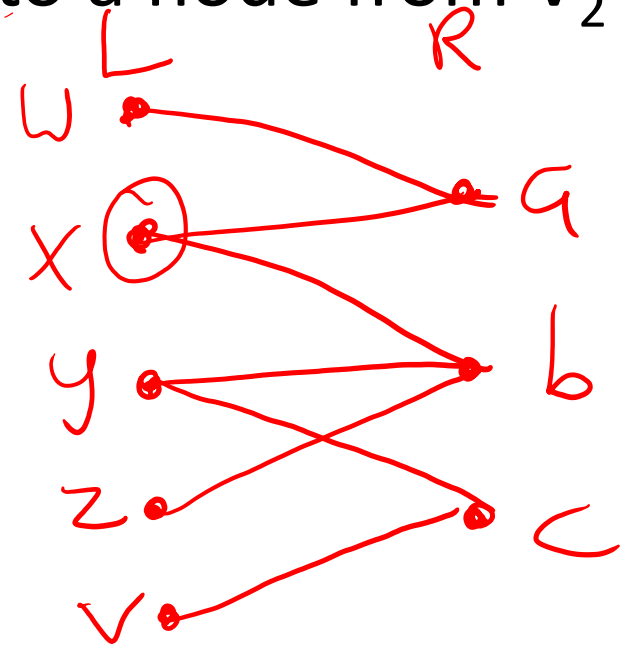
Useful Strategy: Modifying the Input Graph

Bipartite Graphs

- A bipartite graph is a graph in which the nodes can be divided into two sets V_1, V_2, and all edges connect a node from V_1 to a node from V_2
- Examples?

person-product purchasing

person-movies watching graph



L - R - L - R - ...

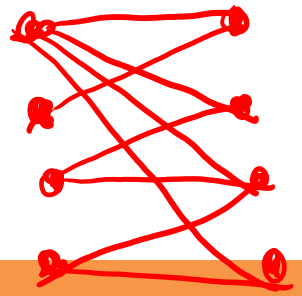
R - L - R - L ...

Bipartite Matching

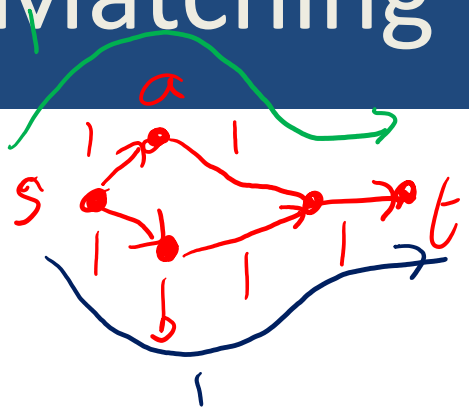
- A bipartite graph is a graph in which the nodes can be divided into two sets V_1, V_2, and all edges connect a node from V_1 to a node from V_2
- A complete matching on a bipartite graph is a set of edges such that every node is adjacent to exactly one edge
- Examples?

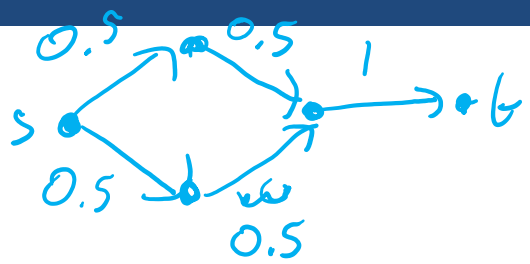
$|L| = |R|$

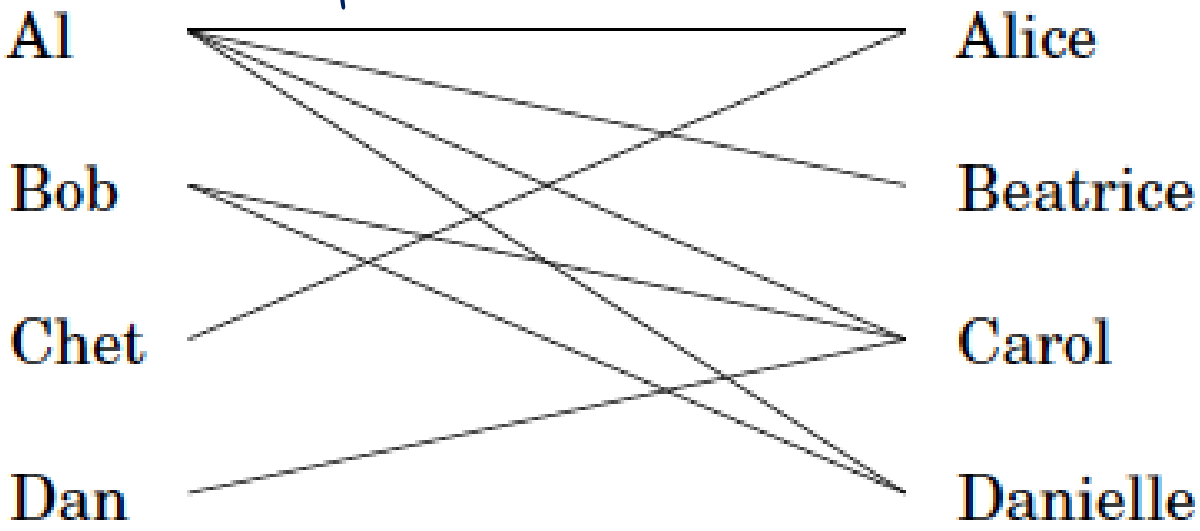
people-vaccine graph



Bipartite Matching

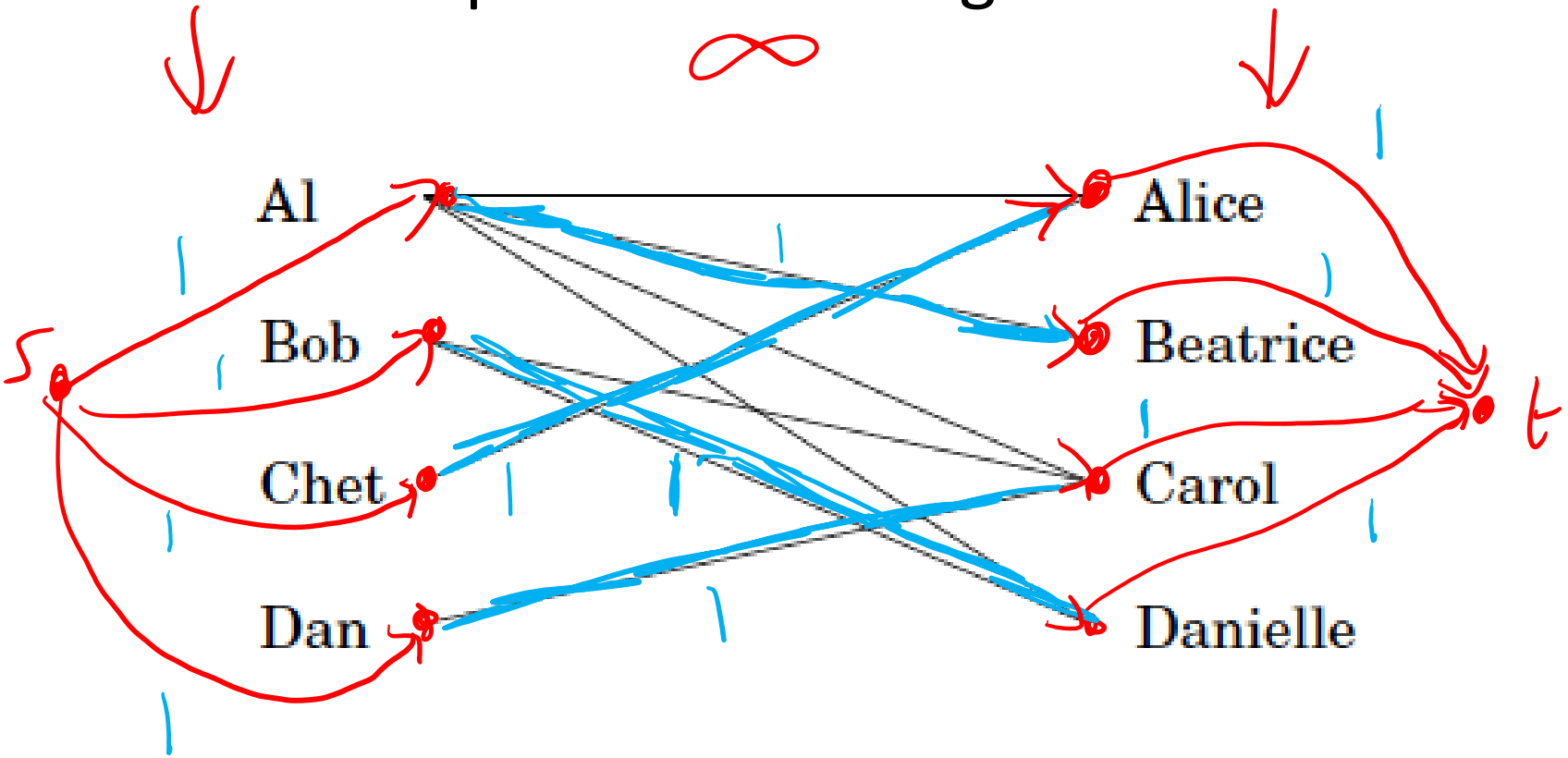






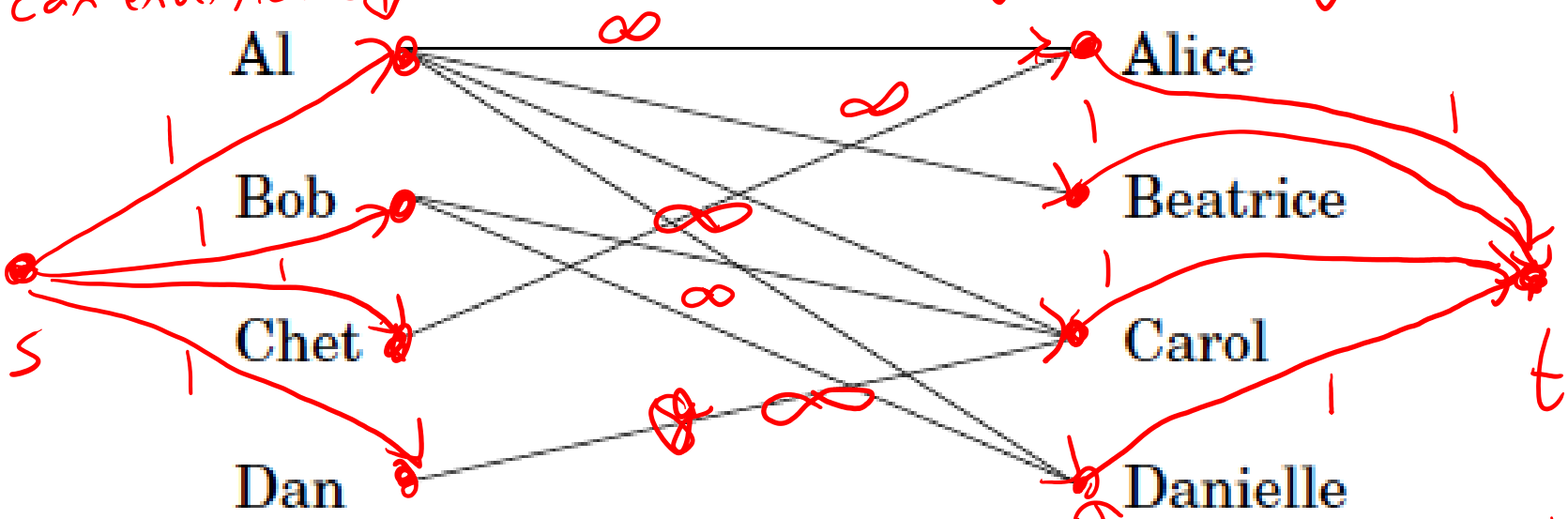
Does a Bipartite Matching Exist?

- How can we use the flow algorithm to determine whether a bipartite matching exists?



Does a Bipartite Matching Exist?

- Useful property: If all edges have integer capacities, then the optimal flow is integral!



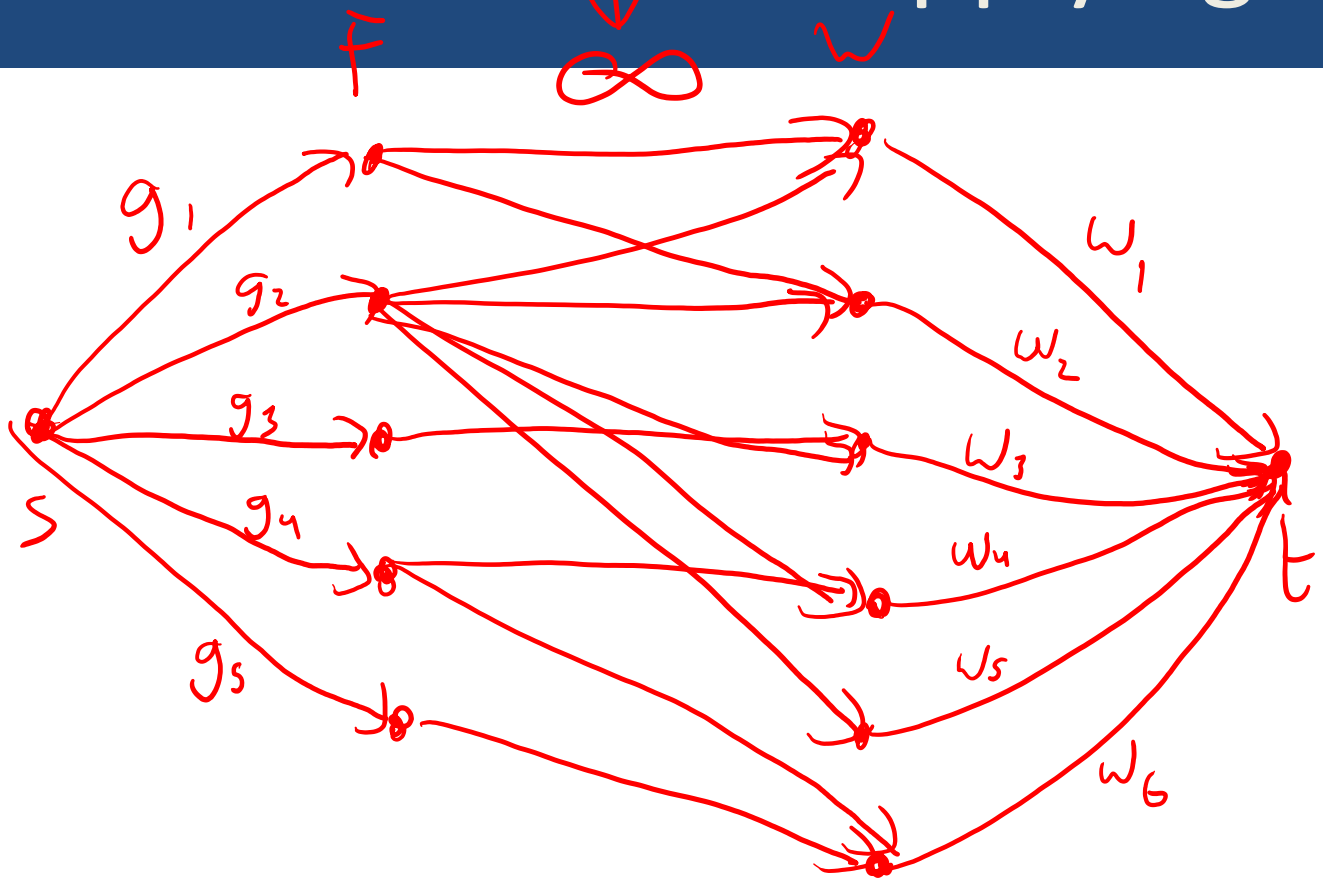
In-Class Exercise

- Come up with a simple algorithm to determine whether a graph is bipartite

In-Class Exercise: Supplying Goods

- There are k factories that produce up to $g_1, \ldots, g_k$ quantity of goods (assume they are integers)
- There are n warehouses that can store up to $w_1, \ldots, w_n$ quantity of goods (assume integers)
- A factory can supply a warehouse if they are within 100 miles of each other (given locations)
- What is the maximum number of goods that can be stored? and how much should each factory produce?

In-Class Exercise: Supplying Goods



factory i can produce $\leq g_i$ units

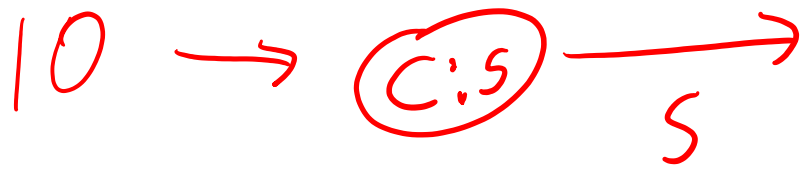
↑ based on dist. constraint

In-Class Exercise: Nodes with Demands

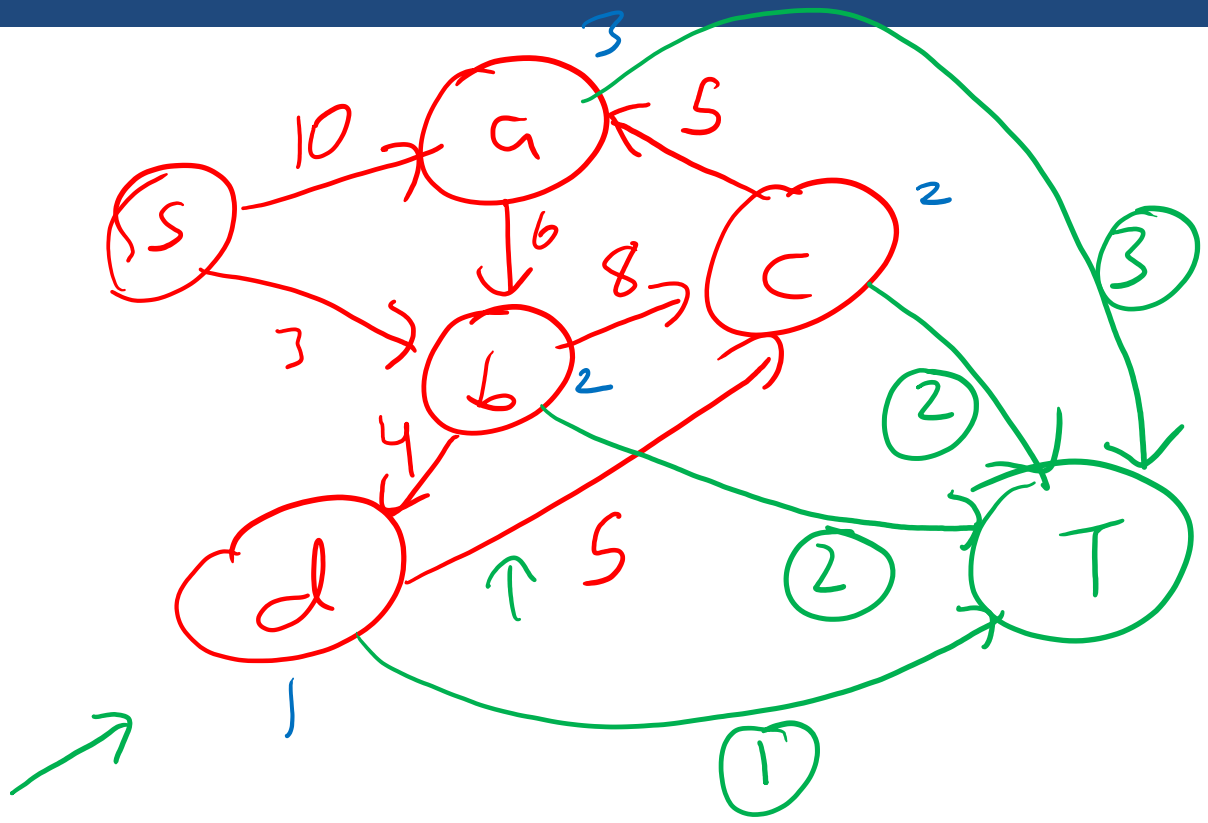
- Suppose that in addition to a maximum capacity on edges, each node has a demand: a certain amount of flow that it will use up
- How do we find if there is a feasible flow?

does a flow exists that satisfies all demands

flow in = flow out + demand



In-Class Exercise: Nodes with Demands



run F-F from $s \to T$

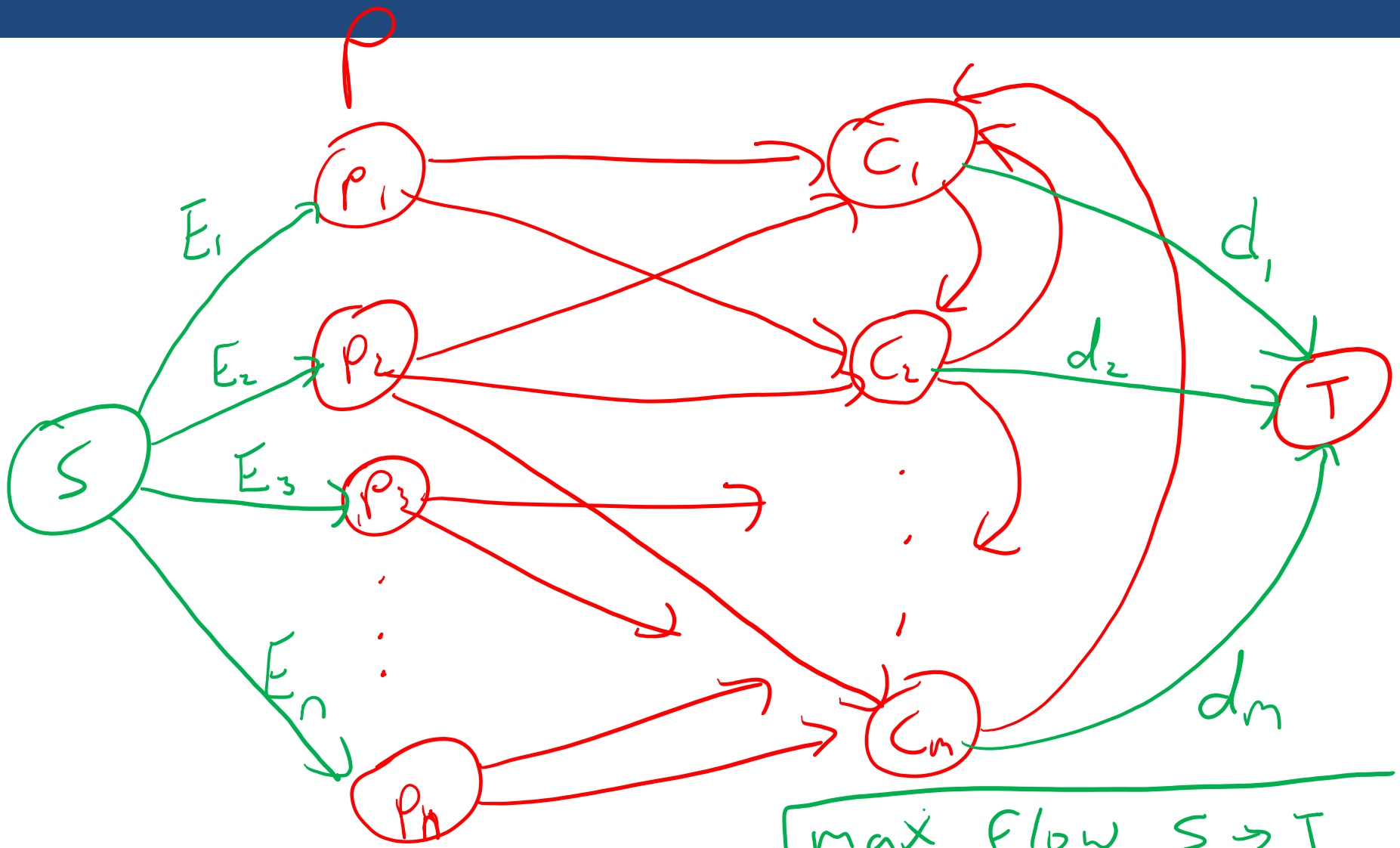
if we can send $1+2+2+3 = 8$ units of flow to T, a feas. flow exists

In-Class Exercise: Powering Cities

- There are several power stations P_1, …, P_n, each generating E_1, …, E_n units of electricity every day
- There are cities C_1, …, C_m, which need D_1, …, D_m units of electricity every day
- Each city is linked to every other city and power station within 100 miles
- Electricity can flow from the power station to a city, and the city can send excess electricity to other cities
- Is it possible to satisfy the demand from every city?

Feasible flow

In-Class Exercise: Powering Cities



In-Class Exercise: Dinner Party

- A group of students is having a dinner party
- You have to decide where people sit
- To help people meet new friends, you decide that no table should include two people from the same major
- There are p majors, with $s_1, \ldots, s_p$ students from each major at the party
- There are q tables, with sizes $t_1, \ldots, t_q$

In-Class Exercise: Dinner Party