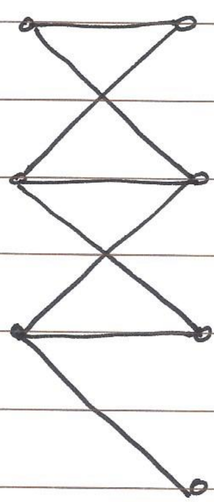
Network Flow

Bipartite Matching Problem

Def. A bipartite graph $G=(V,E)$ is an undirected graph whose node set can be partitioned as $V = X \cup Y$ with property that every edge $e \in E$ has one end in X and the other in Y.

Def. A matching M in G is a subset of the edges $M \subseteq E$ such that each node appears in at most one edge in M.



G

Problem Statement:

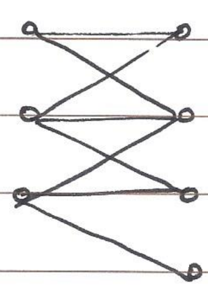
Find a matching M of largest possible size in G.

General Plan:

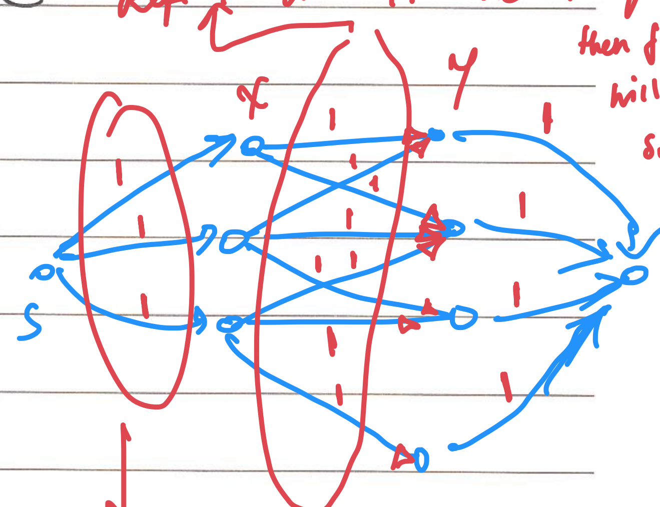
Design a flow network G' that will have a flow value $v(f) = k$ iff there is a ~~max.~~ ~~size~~ matching of size k in G. Moreover, flow f in G' should identify the matching M in G.

Construction of G'

capacity of edges should be kept 1 as if we keep suppose 2 then flow through it will be 0, 1 or 2 so this if flow=1 through edge we can't judge if it is included or not In 0, 1 capacity it is clear



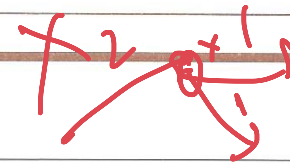
G



Should be 1

G'

Value of flow(G') gives largest matching in G



but we want node to be in atmost 1 edge

Solution

Run Max Flow on G'. Say max. flow is f.

Edges carrying flow between sets X & Y will correspond to our max. size matching in G.

To prove this, we will show that G' will have a max. flow of value $\underline{k}$ iff G has a max. size matching of size $\underline{k}$.

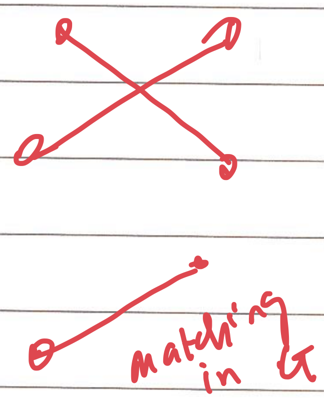
Proofs always have 2 components

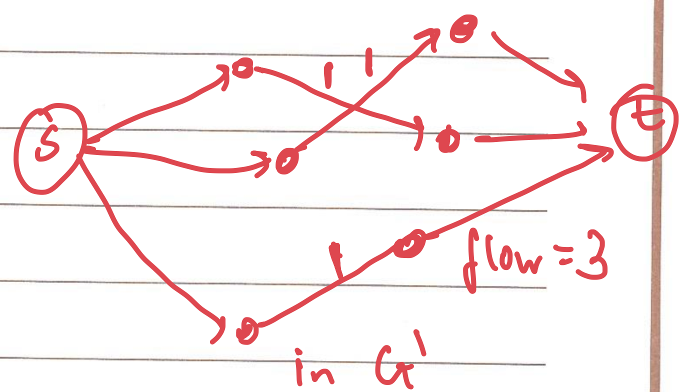
1. If you give me soln to G, I can find same flow in G'

2. For flow f in G', there is a matching of same size in G

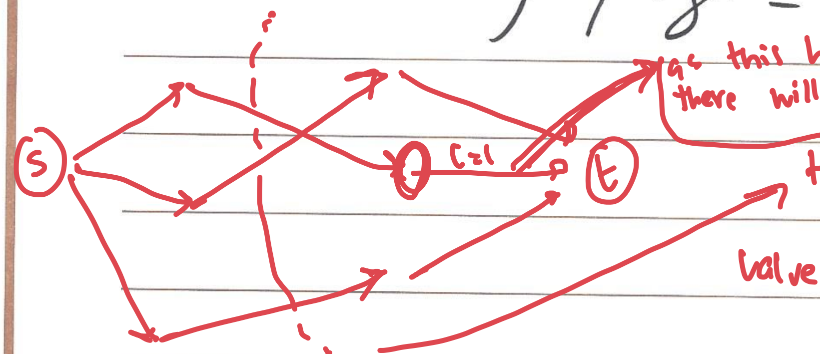
Proof:

A- If we have a matching of size $\underline{k}$ in G, we can find an s-t flow f of value $\underline{k}$ in G'.





B- If we have an s-t flow of value $\underline{k}$ in G', we can find a matching of size $\underline{k}$ in G.



as this has capacity of 1 there will be exactly 1 edge carrying flow to this edge or else c≥2

this cut

value of flow = c(A, B)

= k * 1

All edges must carry flow.

Given flow 'k' in G'

Using Ford Fulkerson to solve max flow problem in G'

$$O(\underbrace{C \cdot m}_{}) =$$

$$C = \sum_{\text{out of } s} c_e$$

all have capacity = 1

$\therefore$ $O(n \cdot m)$ → no. of nodes

Network Flow

Edge-Disjoint Paths

Def. A set of paths is edge-disjoint if their edge sets are disjoint

Problem Statement

Given a directed graph G with $s \& t \in V$, find max. number of edge-disjoint s-t paths in G.

each edge can only be used in 1 path.

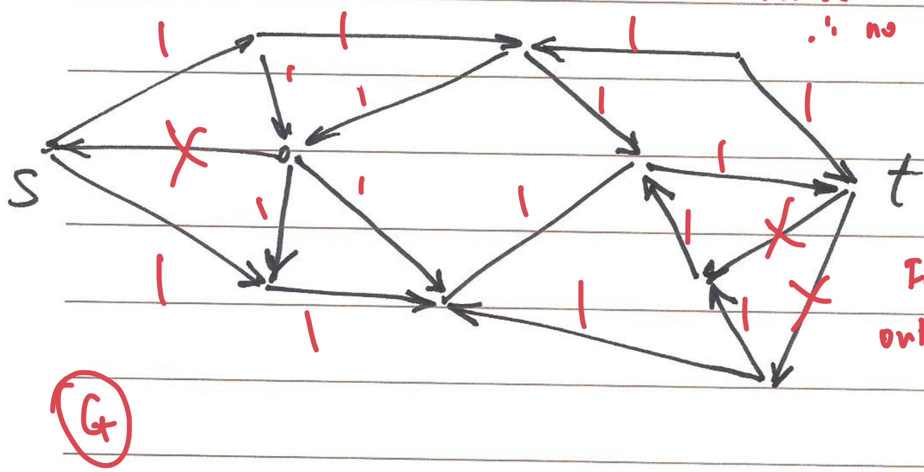
General Plan:

Design a flow network G' that will have a flow value $v(f) = k$ iff there are $\underline{k}$ edge-disjoint s-t paths in G. Moreover, flow f in G' should identify the set of edge-disjoint paths in G.

Construction of G'

Source should be proper source

∴ no incoming edges



For sink no outgoing edge

Set all edge capacities to 1

run max flow.

Solution:

. Run Max flow in G'

. v(f) will equal the max. number of edge-disjoint s-t paths

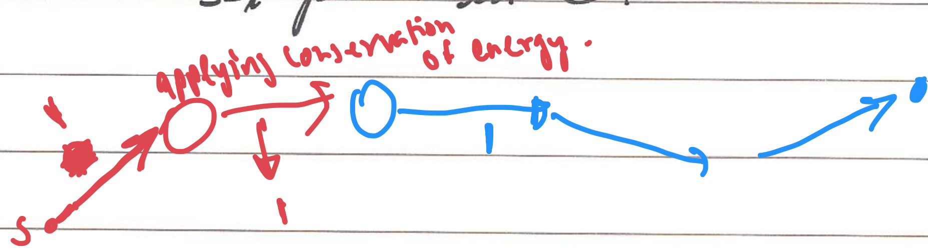
. f will identify edges on these paths

To prove this, we will show that there are $\underline{k}$ edge disjoint paths in G iff there is a flow of value $\underline{k}$ in G'!!

Proof:

A) If we have $\underline{k}$ edge disjoint s-t paths in G, we can find a flow of value $\underline{k}$ in G'.

B) If we have a flow of value $\underline{k}$ in G', we can find $\underline{k}$ edge-disjoint s-t paths in G.



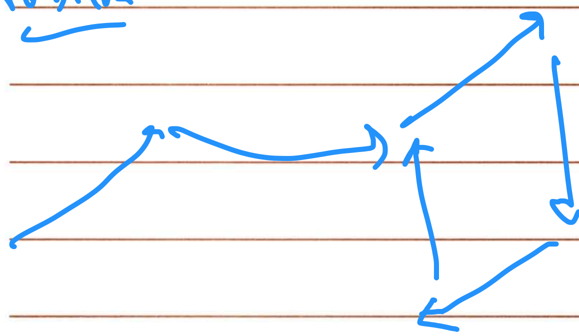
if $v(f) = k > 0$

$\therefore$ there is atleast one edge out from s carrying flow

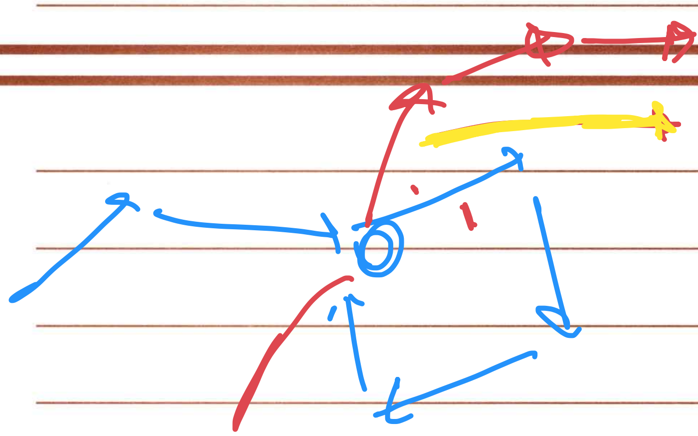
after reaching 't' we have flow=1

$\therefore$ for rest k-1 flow we will do same thing.

problem

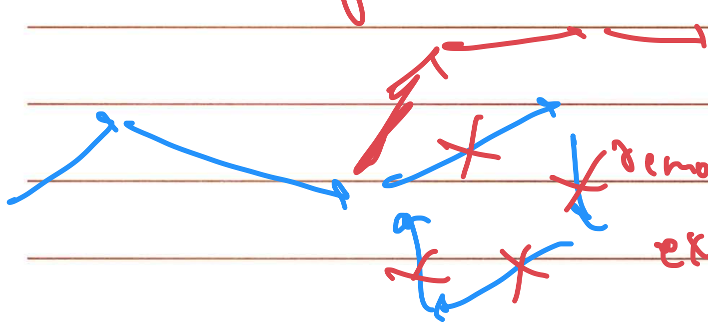


While exploring a path I land on a particular node again ∴ cycle



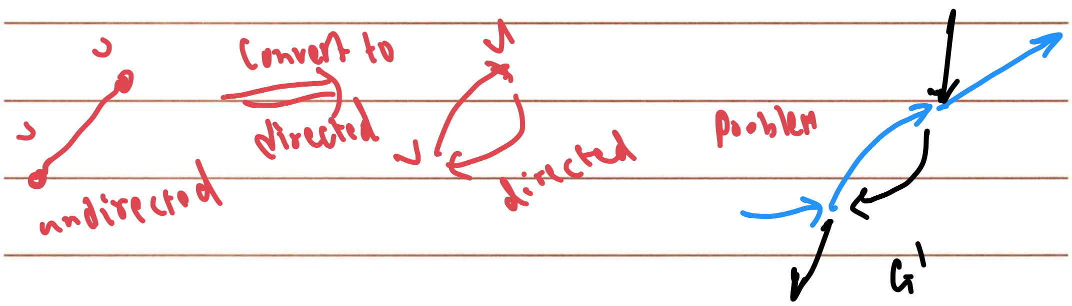
∴ there is some other edge from this edge to ensure conservation of energy.

Incoming flow = 2

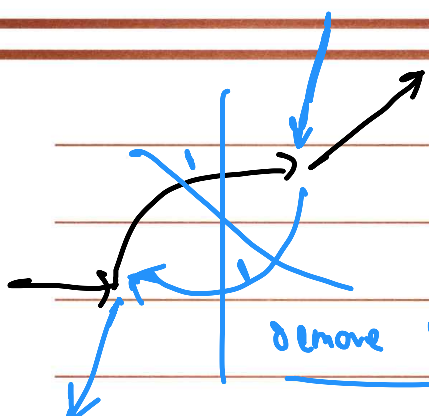


remove cycle and explore the other node

How do we modify this solution so it applies to undirected graphs?



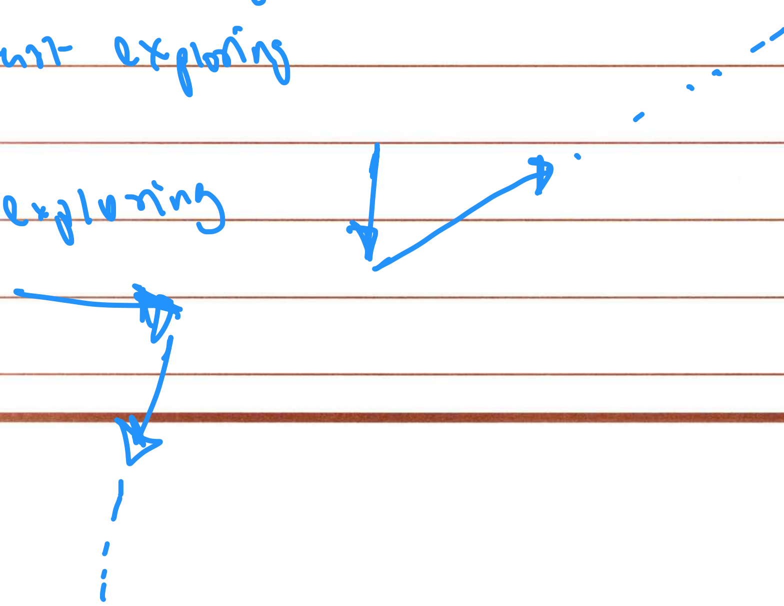
Same edge is used in two paths
so not edge disjoint



Remove cycle

This removing is done before we start exploring

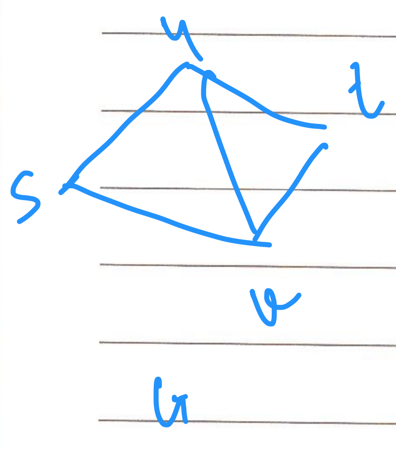
So while exploring

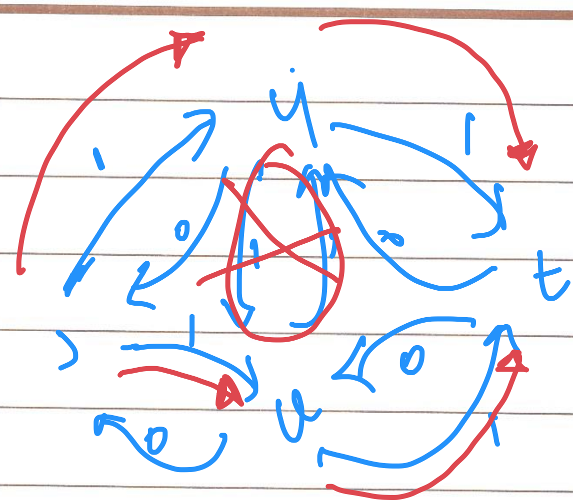


Network Flow

Node-disjoint Paths

example of above





∴ only remove cycles corresponding to one edge one (u,v) edge has a cycle so remove

Using Ford Fulkerson $O(c \cdot m)$

$O(n \cdot m)$

Node Disjoint path

Def. A set of paths is node-disjoint if their node sets (except for starting & ending nodes) are disjoint

Problem Statement

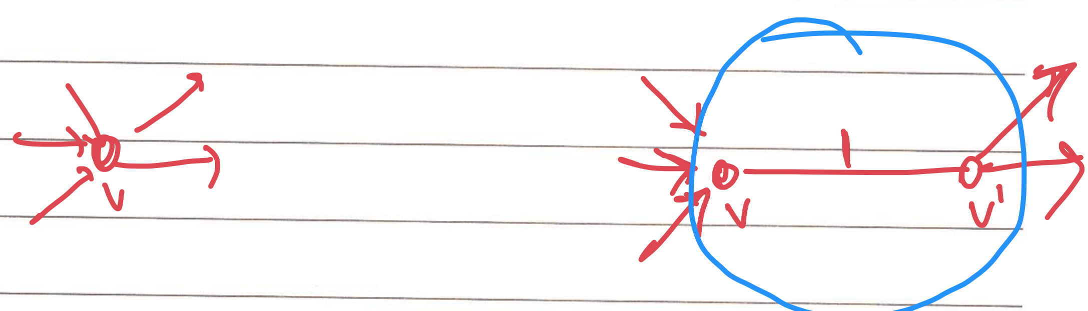
Given a directected graph G with s&t ∈ V, find the max. number of node-disjoint s-t paths in G.

Now here we have to control flow through node

Plan: As usual...

Construction of G':

a node in G ⟶ node in G'



this is our actual node

adding edge 1 we can control flow through node v

Rest is same

new type of network

Circulation &

Circulation with Lower Bounds

Circulation Network

. We are given a directed graph $G=(V,E)$ with capacities on the edges.

. Associated with each node $v \in V$ is a demand d_v

- if $d_v > 0$, node v has demand of d_v for flow (Sink)
- if $d_v < 0$, node v has a supply of $|d_v|$ for flow (Source)
- if $d_v = 0$ v is neither a sink nor a source

we can have multiple sink nodes and multiple source nodes

Def. A **circulation** with demand $\{d_v\}$ is a function f that assigns non-negative real numbers to each edge and satisfies:

1) Capacity condition
   for each edge $e \in E$ $0 \leq f(e) \leq C_e$

2) Demand Condition
   for each node $v \in V$,

$$f^{in}(v) - f^{out}(v) = d_v$$

FACT: If there is a feasible circulation w/ demands $\{d_v\}$ then $\sum_v d_v = 0$

Proof: $f^{in}(v) - f^{out}(v) = d_v$

$$\sum_v d_v = \sum_v f^{in}(v) - f^{out}(v)$$

For some edge $v \xrightarrow[f(e)]{e} u$

This edge will be in $f^{out}(v)$ and $f^{in}(u)$

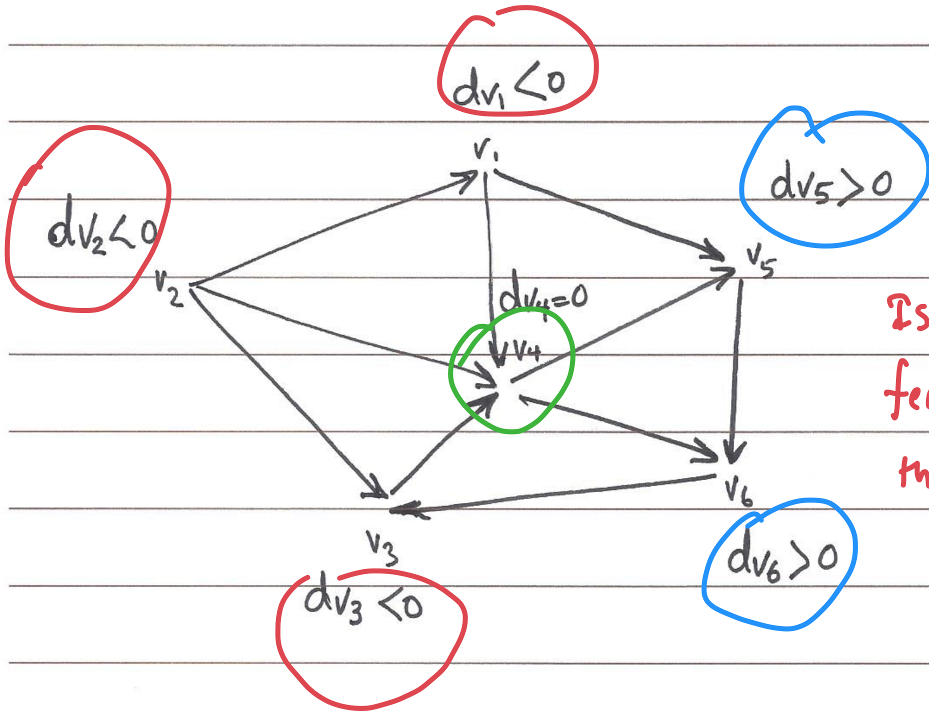
$\therefore$ ~~at~~ each edge is twice with + and -ve values

$$\therefore \sum_v d_v = \sum_v f^{in}(v) - f^{out}(v) = 0$$

$$\sum_{v:\, d_v > 0} d_v = -\sum_{v:\, d_v < 0} d_v = D$$

Total demand value

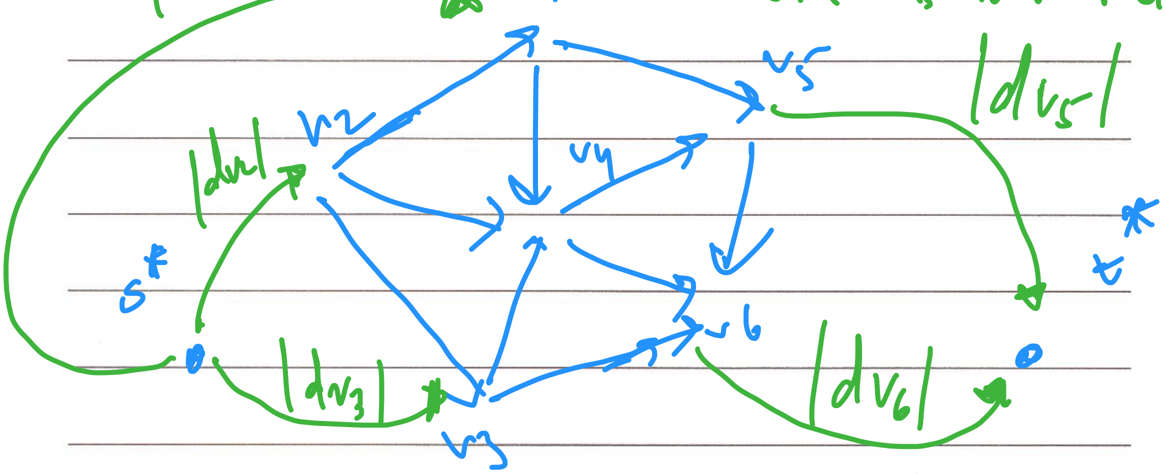
demand = supply



Is there a feasible circulation in this network.

$\underline{G}$

all other capacities are same as that of G



G'

Run max flow on G' $\longrightarrow$ f

say $v(f) = x$

$$D = \sum_{v: d_v > 0} d_v = -\sum_{v: d_v < 0} d_v$$

Can $x < D$?

Yes, if there is no feasible circulation

Can $x > D$?

Not possible

d_{vu}

t^*

d_{vs}

$D = d_{vu} + d_{vs}$

$\therefore$ cut $= D$

Can $x = D$

Yes, this says there is a feasible solution.

$\therefore$ $v(f)$ can never be greater than flow of any cut

Proof:

A) If there is a feasible circulation f w/ demand values $\{d_v\}$ in G, we can find a Max Flow in G' of value D.

B) If there is a Max Flow in G' of value D, we can find a feasible circulation in G.

Circulation with Demands & Lowerbounds

Conditions:

1) capacity condition
   for each edge $e \in E$, $l_e \leq f(e) \leq c_e$ only this is diff

2) Demand condition
   for every node $v \in V$, $f^{in}(v) - f^{out}(v) = d_v$

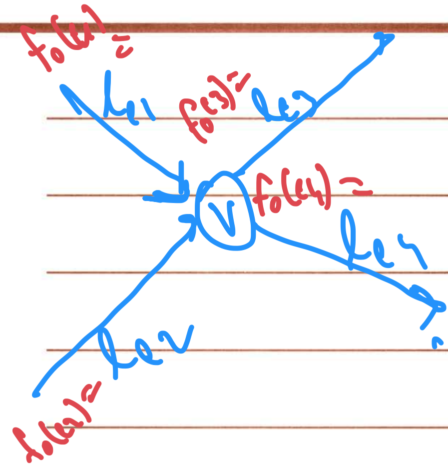
Solution:
Find feasible circulation (if it exists) in two passes.

Pass #1. find f_0 to satisfy all l_e's
removing capacity conditions

Pass #2. Use remaining capacity of the network to find a feasible circulation f_1 (if it exists)

Combine the two flows: $f = f_0 + f_1$

reduce it to circulation problem



L_v = imbalance at node v due to f_0

as there is no conservation of flow in circulation networks

$$f_0^{in}(v) - f_0^{out}(v) = \sum_{e \text{ into } v} l_e - \sum_{e \text{ out of } v} l_e$$

$\therefore$ we use following sol^n.

1. push flow f_0 thore G where $f_0(e) = l_e$

2. Construct G' where $c_e' = c_e - l_e$

$d_v' = d_v - L_v$

3. Find feasible circulation in G'

Call this f'

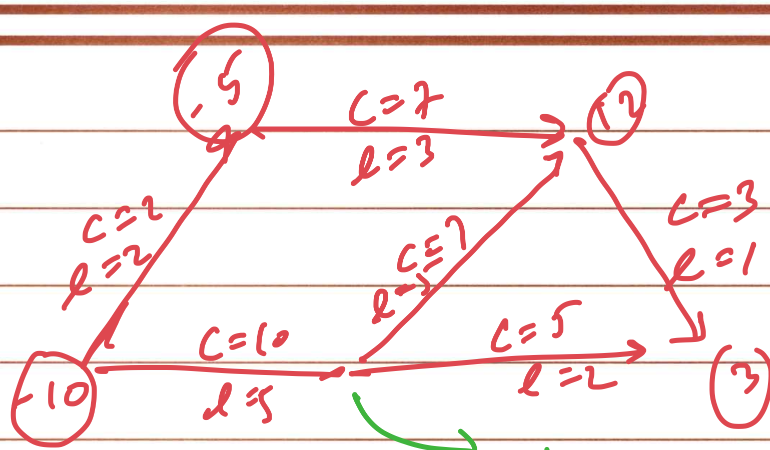
4. If there is no feasible circulation in G'

$\therefore$ no feasible circulation in G

otherwise, feasible circulation in G

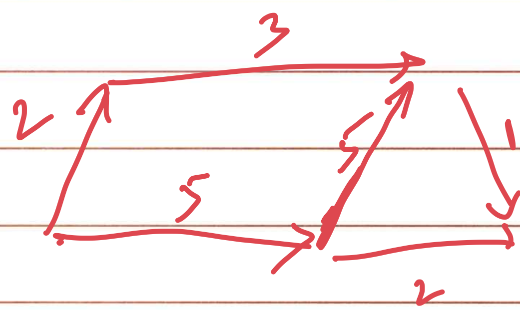
$\Rightarrow f_0 + f_1$

example

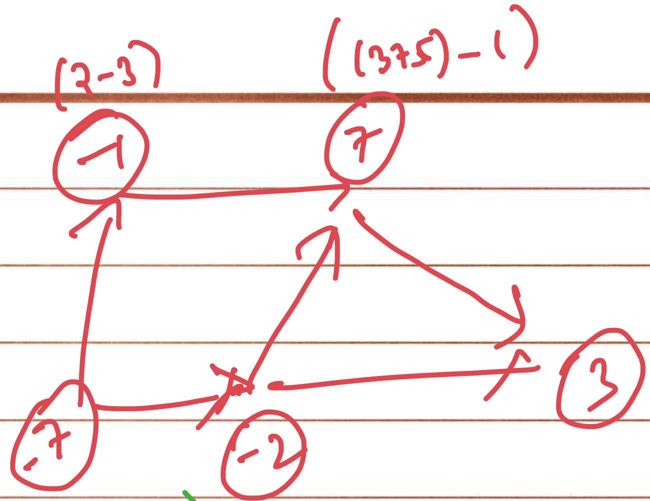


nothing mentioned means demand = 0

f_0

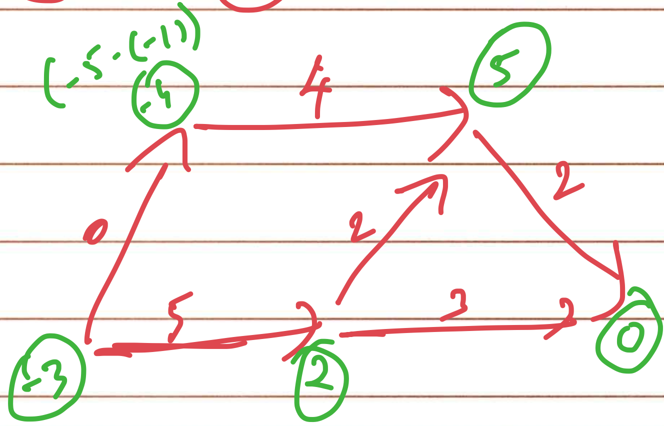


Step 1



Lv

Step2



G'

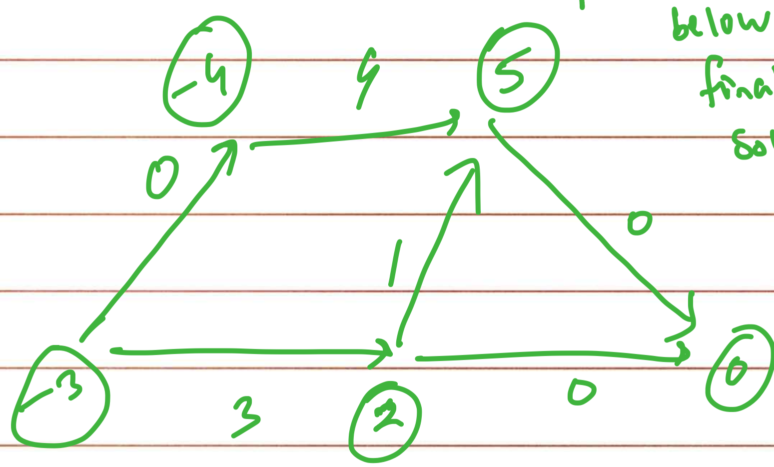
Step3

IF Flow = demand = 7 ⇒ f_1

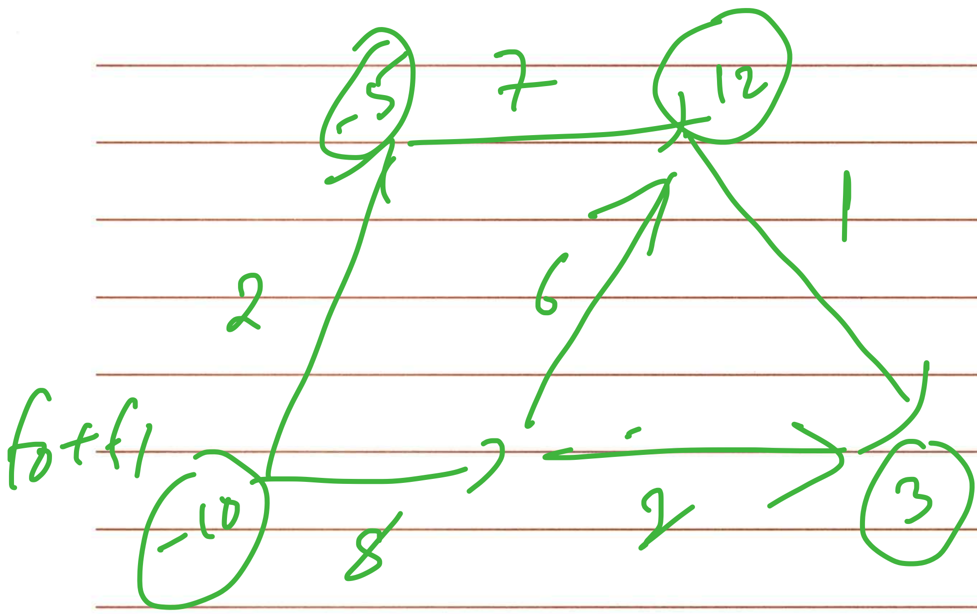
there is a feasible soln.

convert G' to solve max flow problem

below given is final ans after solving max flow



f_1


-5
7
12
2
6
1
f0+f1
-10
8
2
3

Survey Design Problem

Survey Design

Input: - Information on who purchased which products
- Maximum and Minimum number of questions to send to customer i
- Maximum and minimum number of questions to ask about product j

demand = total flow into node − total out of node

- Max flow
- Circulation prob
- Circulation with lower bounds $\boxed{\checkmark}$

we know flow into t = flow out of s

as there is minimum condition on questions

as they will be asked about a product or not asked

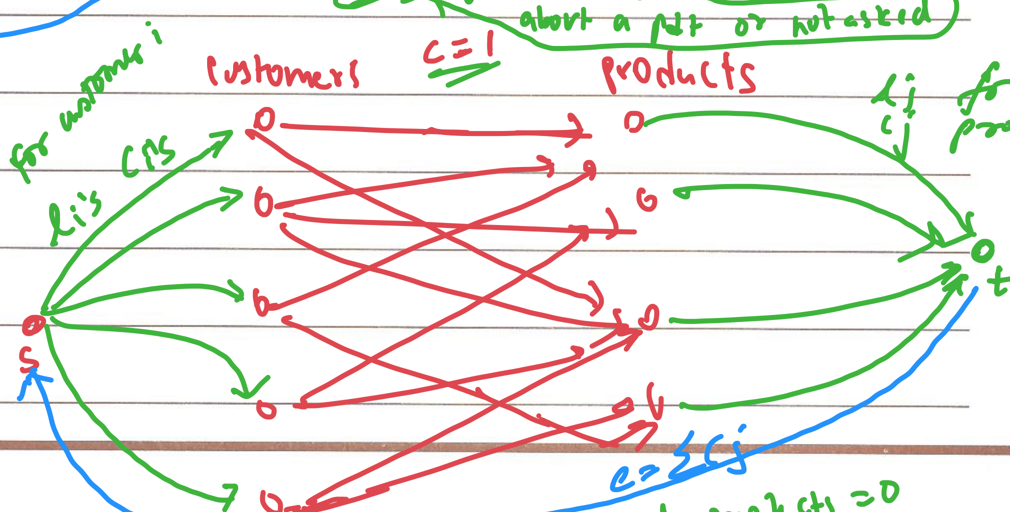
demand value for s and t

for s ~~{ ?~~

~~{ ?~~

no

so what?



so we introduce s t edge

∴ now Demand(s)=0 Demand(t)=0

demand value for customers & products = 0

customer connected to a product if they have purchased it

Min Flow Problem

Input: Directed graph $G=(V,E)$

Source $s \in V$

Sink $t \in V$

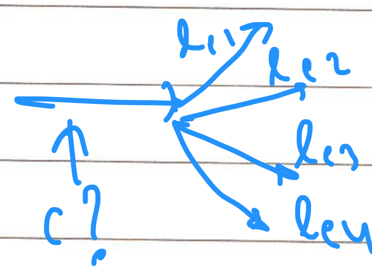
l_e for each edge $e \in E$

There are no c_e's

Objective: Find a feasible flow of minimum possible value

Can be solved in 2 passes

pass 1: satisfy min cap. condition



$c = \sum l_e$'s for all edges

replace capacity with this than ∞

Now find feasible circulation.

Solution:

- Assign "large" capacities to all edges and find a feasible flow f

excess flow going through every edge

- Construct G', where all the edges are reversed and the reversed edge e has capacity $= f_e - l_e$

find maximum of this excess flow

- Find maximum flow from t to s in G'

- Min flow $= f - f'$

min flow = flow from satisfying constraints − max excess flow

Discussion 9

1. We're asked to help the captain of the USC tennis team to arrange a series of matches against UCLA's team. Both teams have *n* players; the tennis rating (a positive number, where a higher number can be interpreted to mean a better player) of the *i*-th member of USC's team is t_i and the tennis rating for the *k*-th member of UCLA's team is b_k. We would like to set up a competition in which each person plays one match against a player from the opposite school. Our goal is to make *as many matches as possible* in which the USC player has a higher tennis rating than his or her opponent. Use network flow to give an algorithm to decide which matches to arrange to achieve this objective.

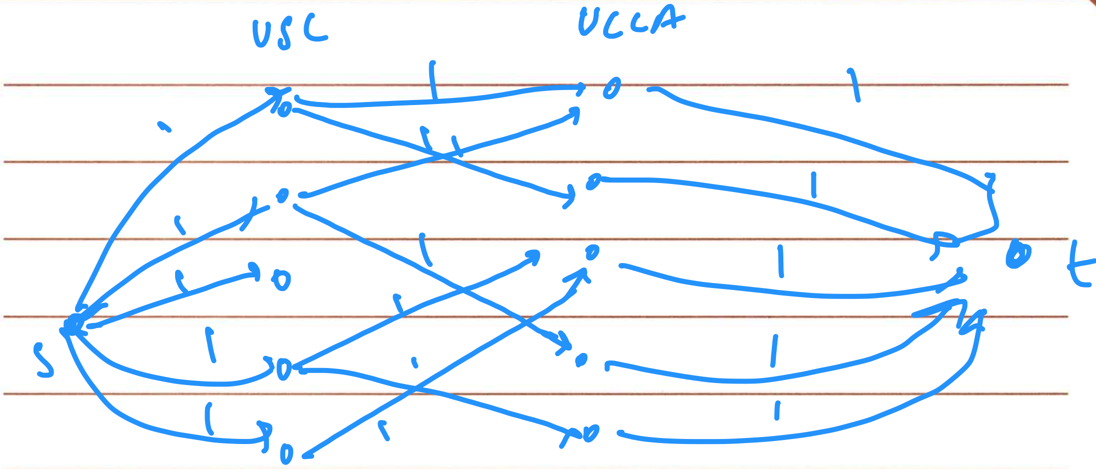
2. CSCI 570 is a large class with *n* TAs. Each week TAs must hold office hours in the TA office room. There is a set of *k* hour-long time intervals $I_1, I_2, \ldots I_k$ in which the office room is available. The room can accommodate up to 3 TAs at any time. Each TA provides a subset of the time intervals he or she can hold office hours with the minimum requirement of l_j hours per week, and the maximum m_j hours per week. Lastly, the total number of office hours held during the week must be *H*. Design an algorithm to determine if there is a valid way to schedule the TA's office hours with respect to these constraints.

3. There are n students in a class. We want to choose a subset of k students as a committee. There has to be m1 number of freshmen, m2 number of sophomores, m3 number of juniors, and m4 number of seniors in the committee. Each student is from one of k departments, where k = m1 +m2 +m3 +m4. Exactly one student from each department has to be chosen for the committee. We are given a list of students, their home departments, and their class (freshman, sophomore, junior, senior). Describe an efficient algorithm based on network flow techniques to select who should be on the committee such that the above constraints are all satisfied.

4. Given a directed graph G=(V,E) a source node s ϵ V, a sink node t ϵ V, and lower bound ℓ_e for flow on each edge e ϵ E, find a feasible s-t flow of minimum possible value.
Note: there are no capacity limits for flow on edges in G.

Min - flow problem

1



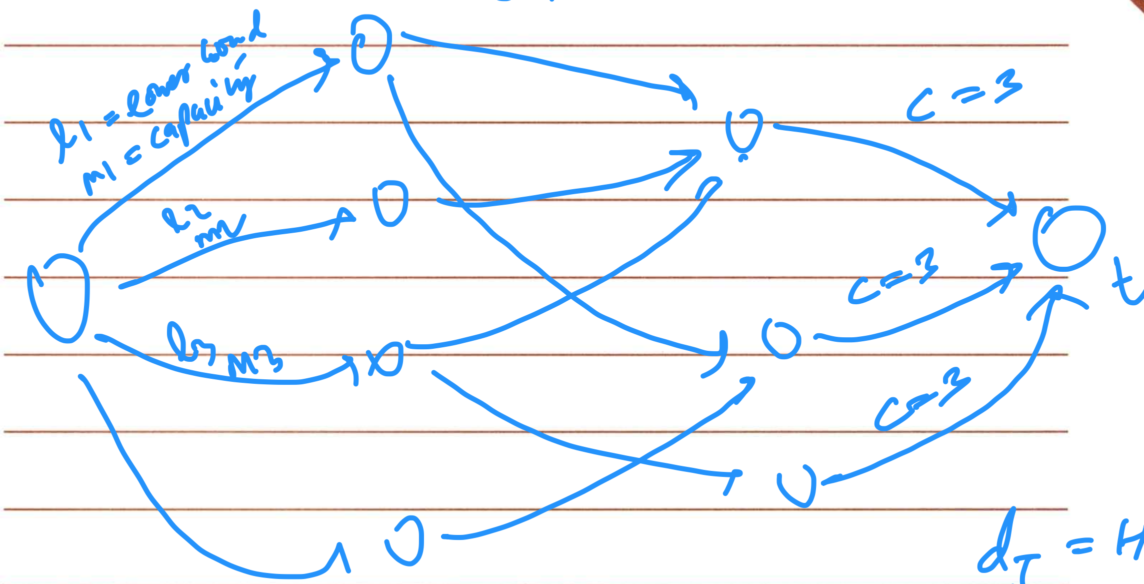
connect if UCLA
player has less rating
than that USC player

Run MAX FLOW.

Suppose $v(f) = x$
This takes care of x games
For other $n-x$ games (randomly allocate)

2

TA   C=1   intervals

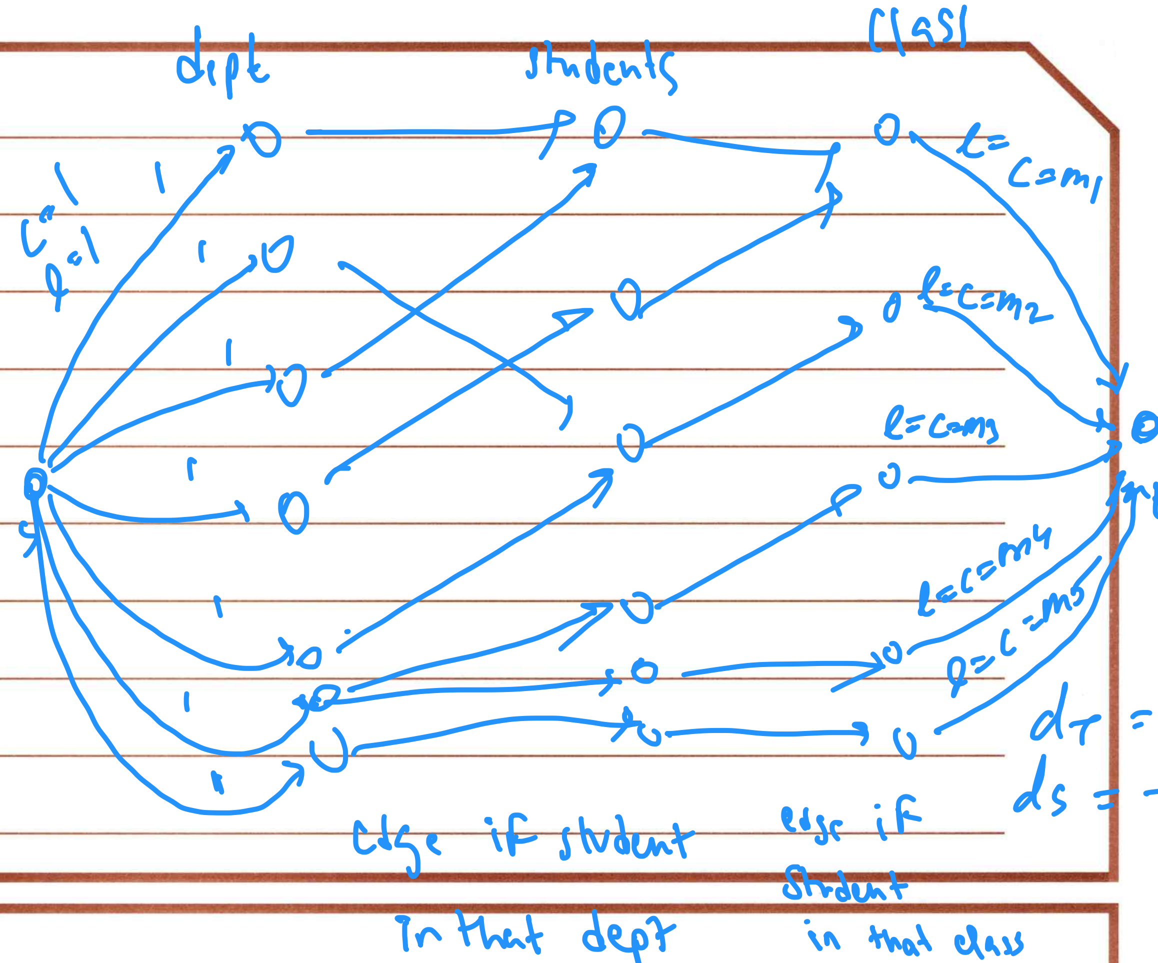


$d_T = H$

$d_S = -H$

if tA is available in that interval

3



$d_T = k$

$d_s = -k$

lower bound here is optional

as here

to satisfy $d_T = k$

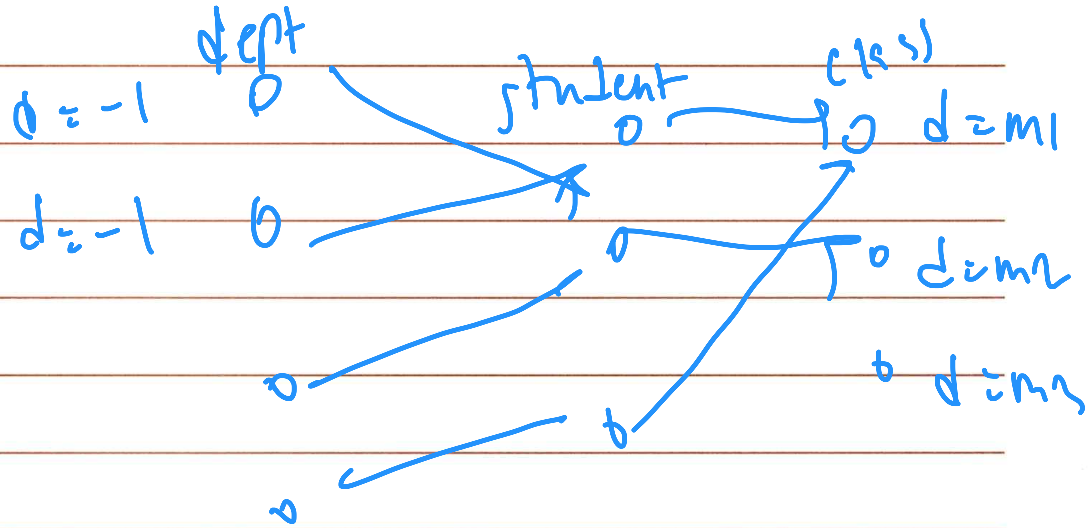
flow through all edges have to be full capacity

we can solve this problem using

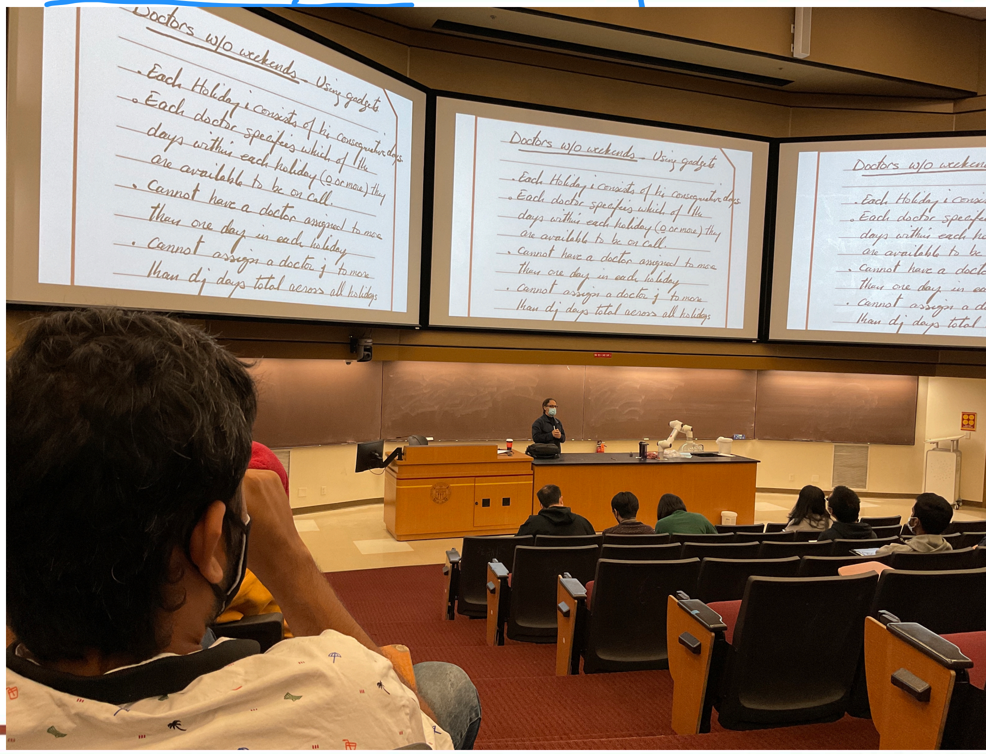
Max flow

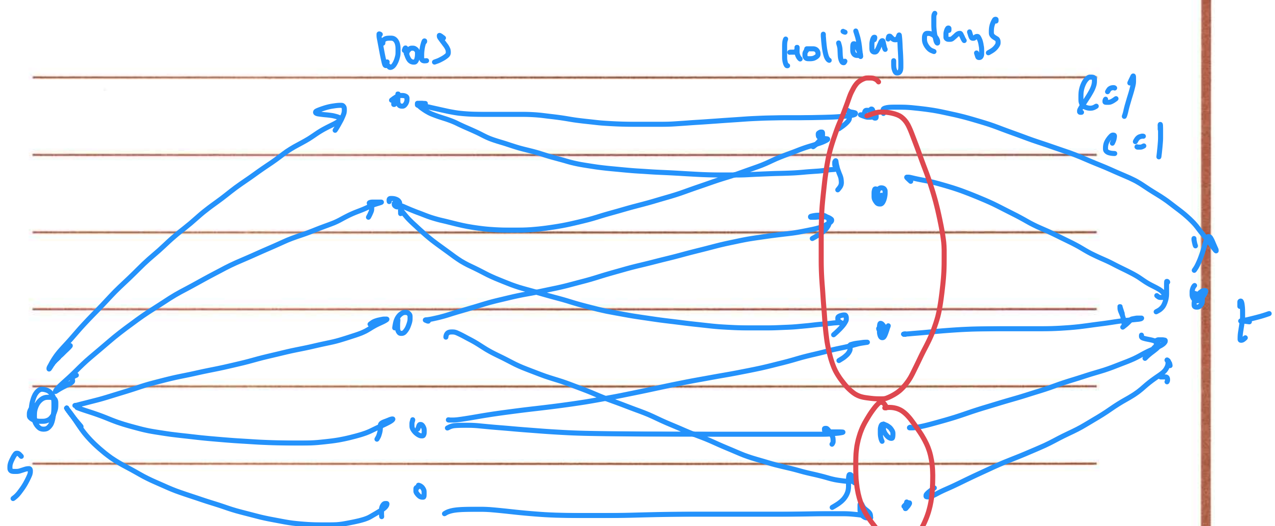
Circulation prob without lower bounds

Solving without S & T



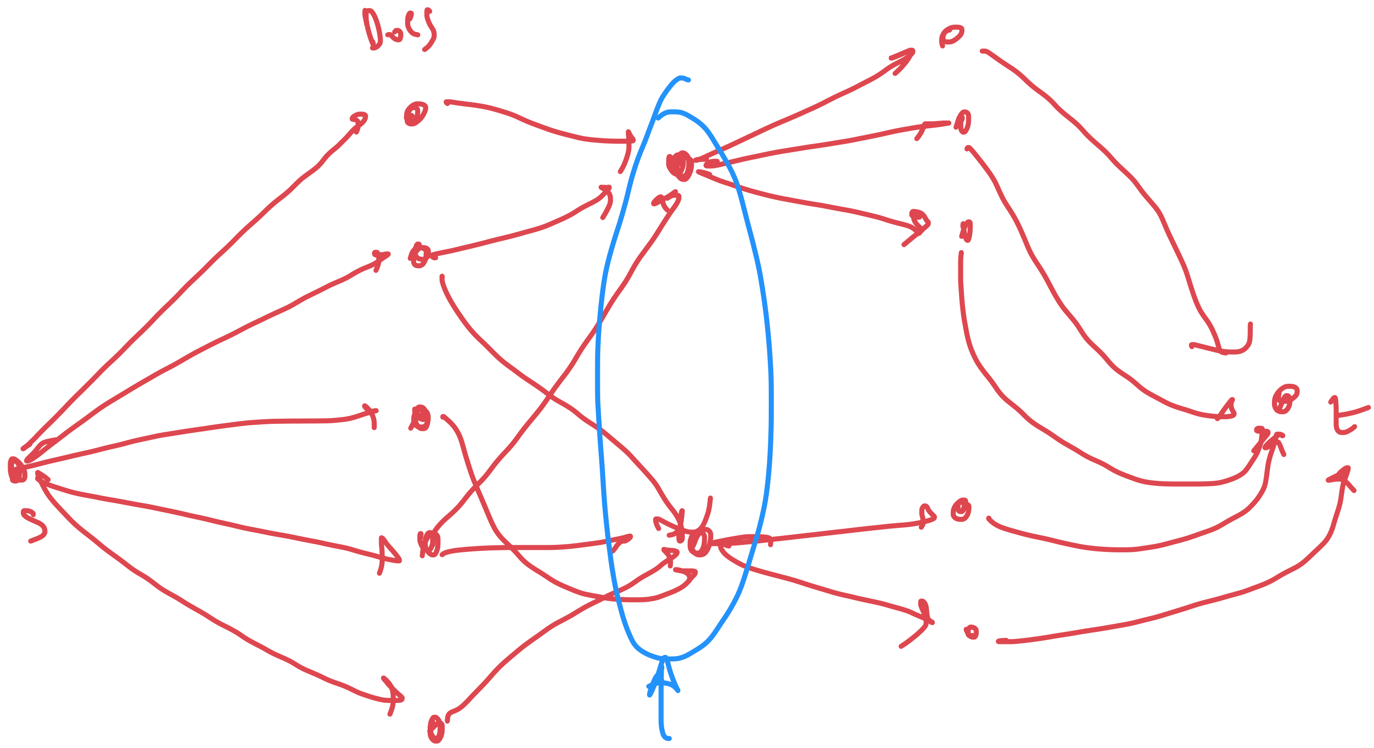
Doctors without weekend problem.





no doc has working day more than 1 in a particular set

These additional graph components are gadgets



These nodes do not correspond to anything

They are just to help us solve the problem

hence they are called gadgets