Analysis of Algorithms

V. Adamchik

CSCI 570

Lecture 9

University of Southern California

Network Flow - 2

Reading: chapter 7

The Ford-Fulkerson Algorithm

$c \in \mathbb{N}^+$

$$|f| = \sum_{\text{out } s} f(e)$$

<u>Algorithm</u>. Given (G, s, t, c)

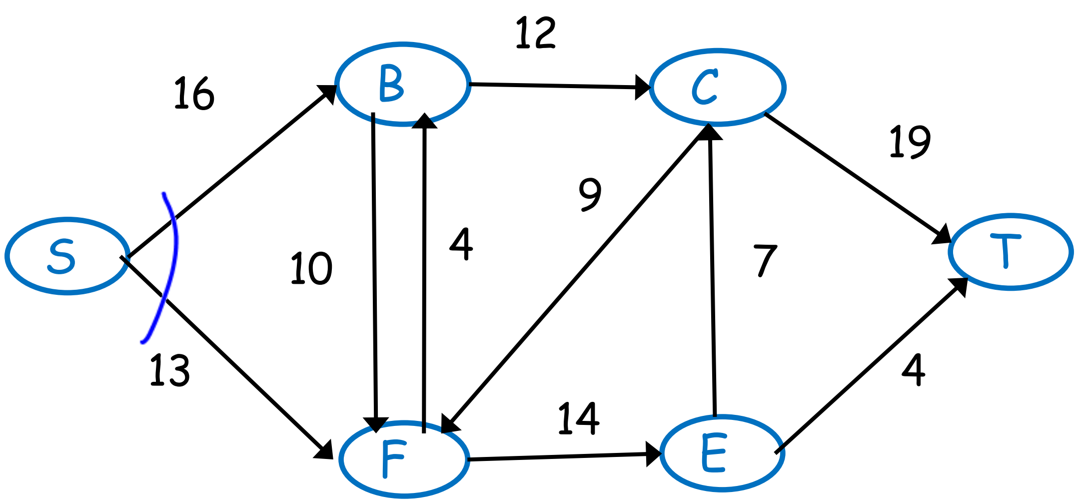
start with f(u,v)=0 and G_f = G.

while exists an augmenting s-t path in G_f

  find a bottleneck

  augment the flow along this path

  update the residual graph G_f



Duality

max-flow	cut
lemma 2	$|f| \le cap(A,B)$
theorem.	$|f| = cap(A,B)$

Reduction

Formally, to reduce a problem Y to a problem X (we write $Y \leq_p X$) we want a function f that maps Y to X such that:

- f is a polynomial time computable
- $\forall$ *instance* $y \in Y$ is solvable if and only if $f(y) \in X$ is solvable.

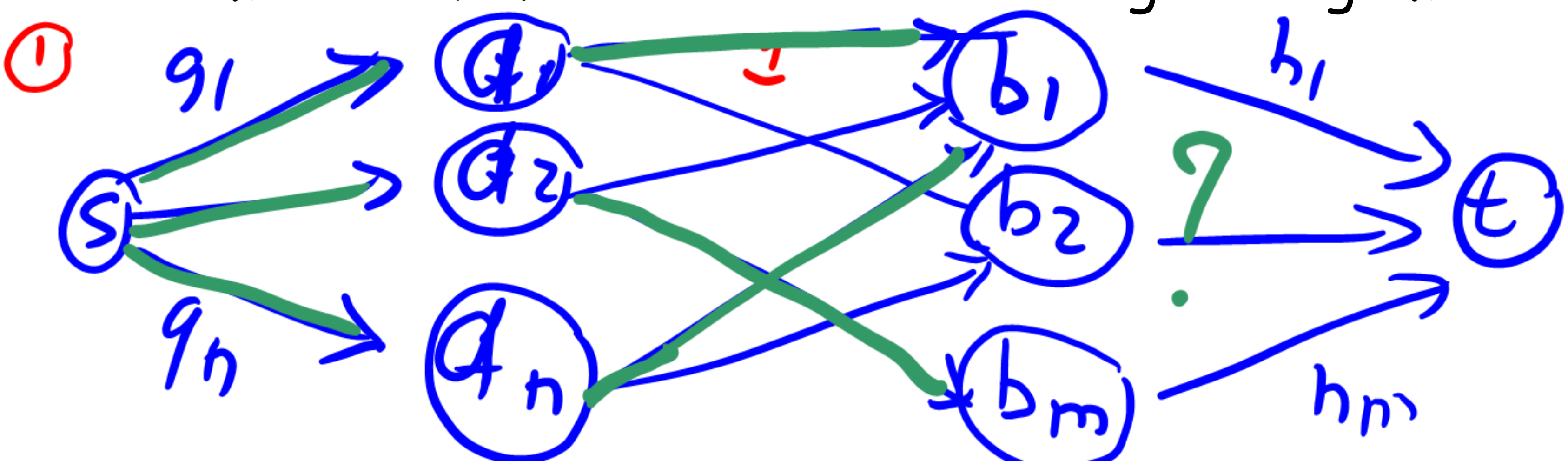
$Y \leq_p NF$

Solving by reduction to NF

1. Describe how to construct a flow network
2. Make a claim. Something like "this problem has a feasible solution if and only if the max flow is ..."
3. Prove the above claim in both directions

Discussion Problem 1

At a dinner party, there are n families $a_1, a_2, \ldots, a_n$ and m tables $b_1, b_2, \ldots, b_m$. The i-th family a_i has g_i members and the j-th table b_j has h_j seats. Everyone is interested in making new friends and the dinner party planner wants to seat people such that no two members of the same family are seated at the same table. Design an algorithm that decides if there exists a seating assignment such that everyone is seated and no two members of the same family are seated at the same table. What would be a seating arrangement?



② Claim. Assignment $\exists$ iff
the max-flow $= g_1 + \dots + g_n$.

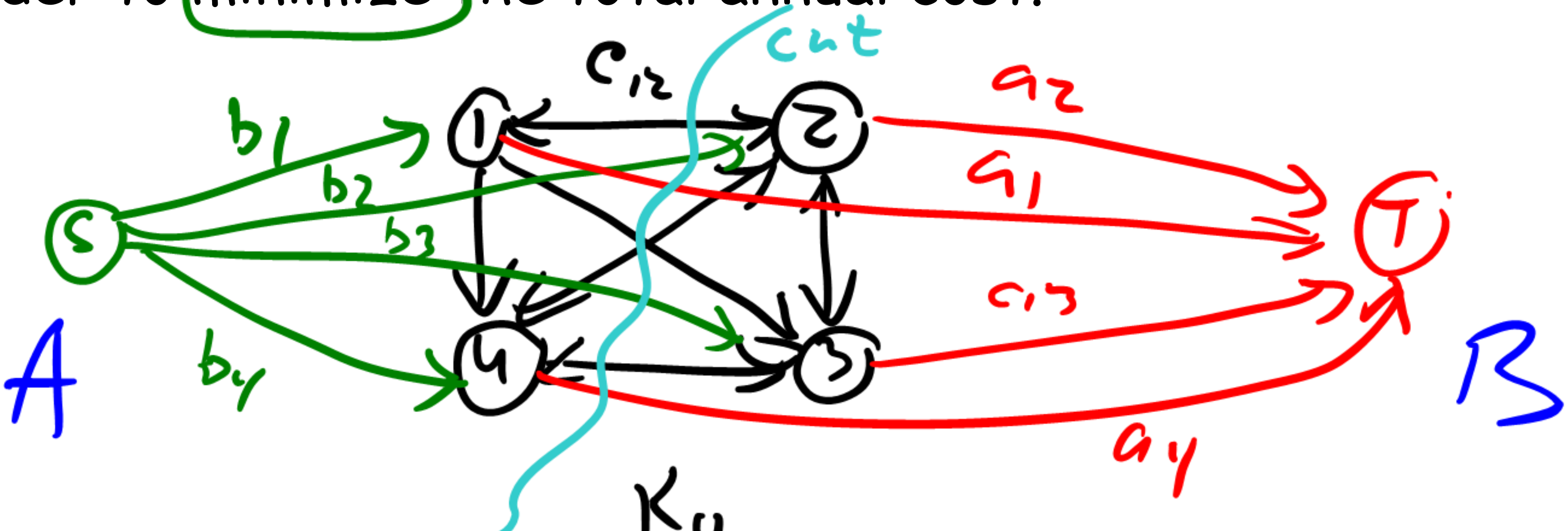
③ $\Rightarrow$ Every member is seated.
We need to prove max-flow $= g_1 + \dots + g_n$

We have a max-flow $= g_1 + \dots + g_n$
$\Leftarrow$ Prove that an assignment $\exists$.

Discussion Problem 2

min-cut

A company has n locations in city A and plans to move some of them (or all) to another city B. The i-th location costs a_i per year if it is in the city A and b_i per year if it is in the city B. The company also needs to pay an extra cost, $c_{ij} > 0$, per year for traveling between locations i and j. We assume that $c_{ij} = c_{ji}$. Design an efficient algorithm to decide which company locations in city A should be moved to city B in order to minimize the total annual cost.



$\mathrm{cap}(A,B) = a_1 + a_4 + b_2 + b_3 + c_{12} + c_{24} + c_{34} + c_{13}$

Goal: min $\mathrm{cap}(A,B)$

Claim. the cost is min iff the max-flow $= \sum_{i \in A} a_i + \sum_{j \in B} b_j + \sum_{i \in A,\, j \in B} c_{ij}$

$\underbrace{}_{a_1+a_2}$ $\underbrace{}_{b_2+b_3}$

Proof.

$\Rightarrow$ construction, min-cut = max-flow

$\Leftarrow$ Given a max-flow.

By property of the min-cut

Runtime. FF: $O(|f| \cdot E)$

$|f| \leq \sum b_i$

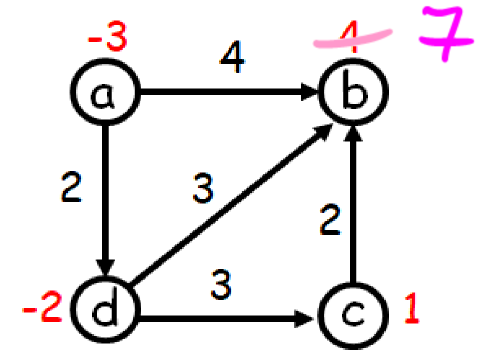
$E = O(n^2)$

Total: $O(|f| \cdot n^2)$

EK: $O(V \cdot E^2)$

Total: $O(n \cdot n^4) = O(n^5)$

Circulation



Given a directed graph in which in addition to having capacities $c(u, v) \geq 0$ on each edge, we associate each vertex v with a supply/demand value $d(v)$. We say that a vertex v is a demand if $d(v) > 0$ and a supply if $d(v) < 0$.

Def. A circulation with demands/supply

$$f : E \to R^+$$

flow $d(v)=0$

1) $0 \leq f(e) \leq c(e)$

2) $f^{in}(v) - f^{out}(v) = \underbrace{d(v)}_{\text{how-!!...}}$

Necessary Condition

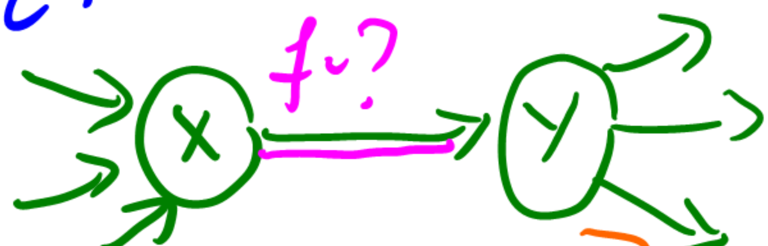
$\exists$ circulation $\Longrightarrow \sum d = 0$

For every feasible circulation $\sum_{v\in V} d(v) = 0$

Proof. $f^{in}(v) - f^{out}(v) = d(v)$ $\left(\begin{matrix} >0 \\ <0 \\ =0 \end{matrix}\right)$

sum up over all vertices

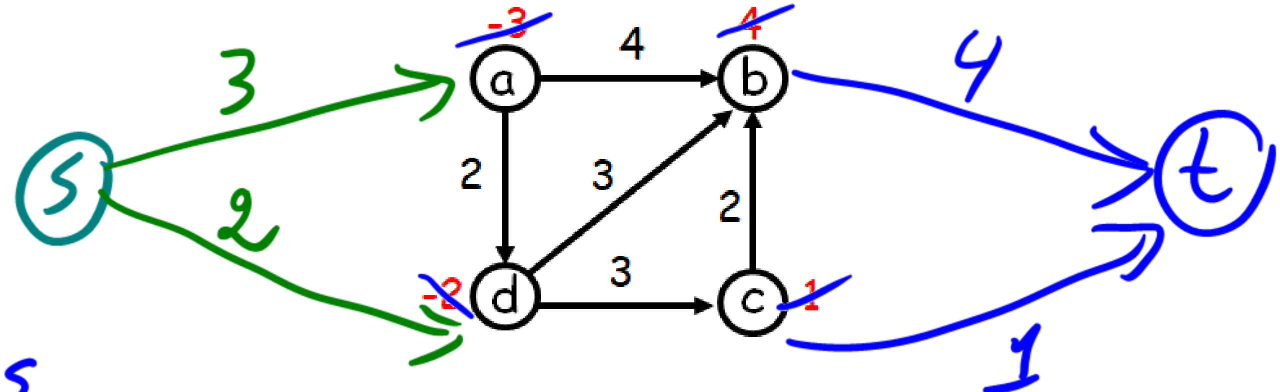
$$0 = \sum_{v \in V} \left[f^{in}(v) - f^{out}(v) \right] = \sum_{v \in V} d(v)$$



$$f^{in}(x) \underbrace{- f^{out}(x) + f^{in}(y)}_{=0} - f^{out}(y)$$

v = x v = y

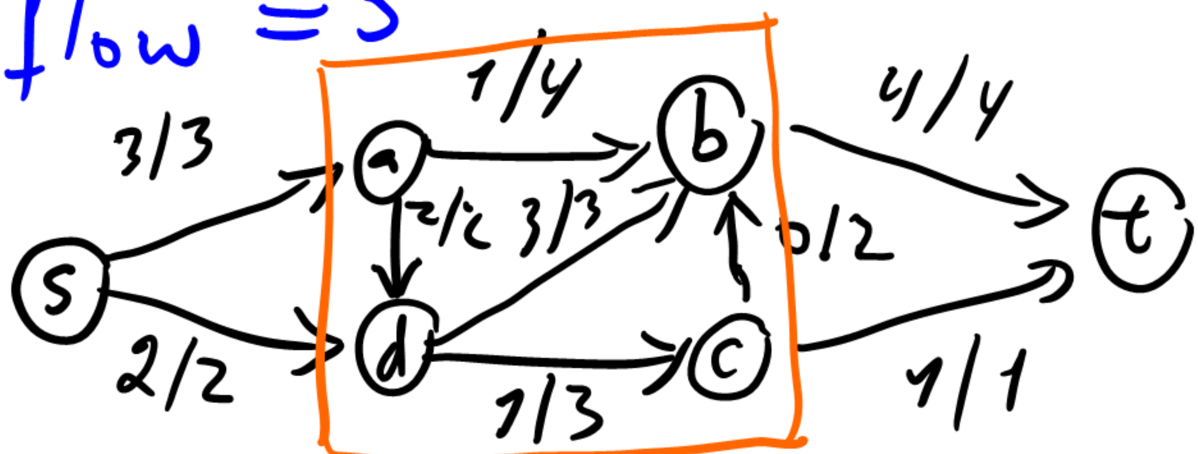
Reduction to Flow Problem



Steps

1. Create NF
2. Run FF or ∀ NF algorithm
3. max-flow = 5

circulation



Circulation with Demands

Claim.

There is a feasible circulation with demands d(v) in G if and only if the maximum s-t flow in G' has value D.

$$\sum_{d(v)>0} d(v) = D$$

Proof.

$\Rightarrow$ $\exists$ circulation by construction

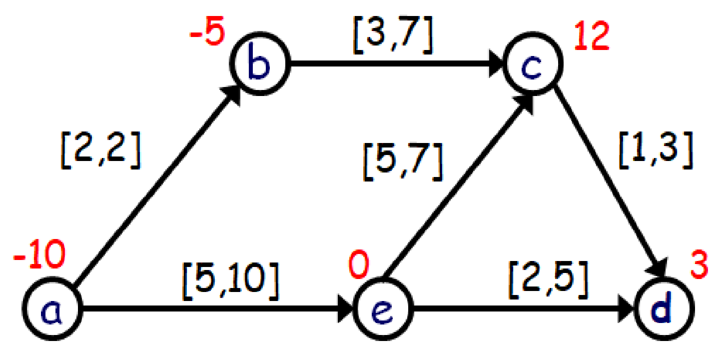
$\Leftarrow$ $\text{max-flow} = \sum_{d(v)>0} d(v)$

We need to find a circulation.

Circulation with Demands and Lower Bounds

We are given a directed graph G=(V, E) with a capacity c(e) and a lower bound $0 \le \ell(e) \le c(e)$ on each edge and a demand d(v) on each vertex.

$[\ell(e), c(e)]$



Def. A circulation

1) $\ell(e) \le f(e) \le c(e)$

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

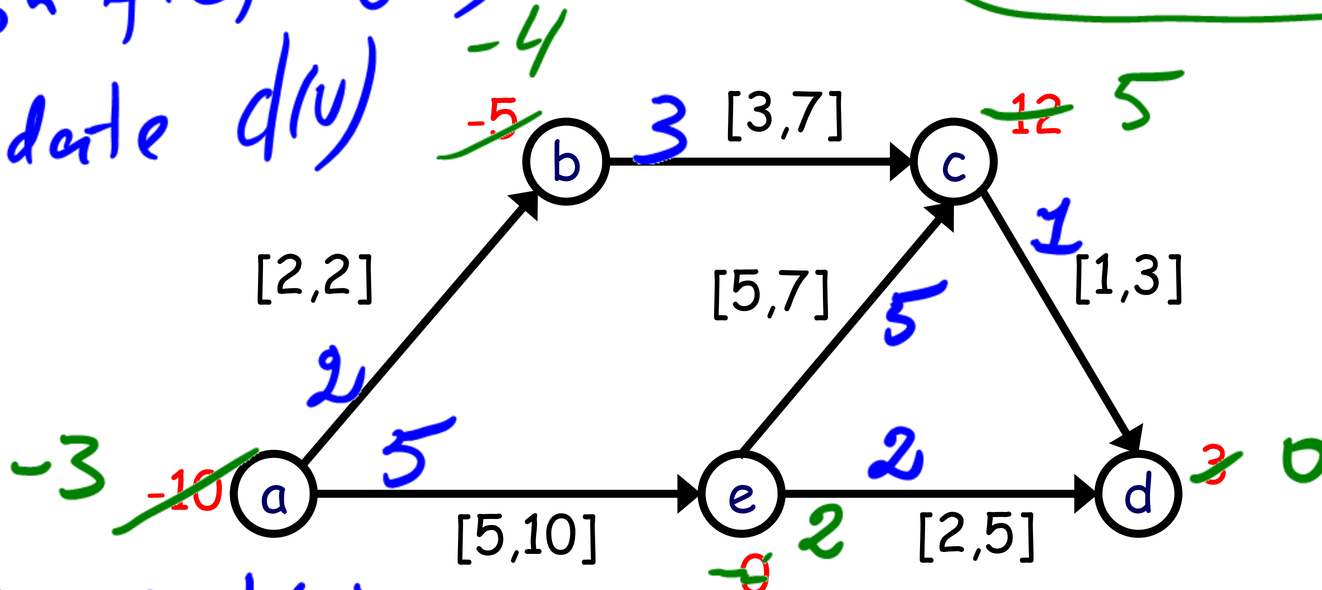
Circulation with Demands and Lower Bounds

Remove lower bounds!

1) Push $f(e) = l(e)$

2) Update $d(v)$

$$L(v) = f_0^{in}(v) - f_0^{out}(v)$$
$$d'(v) = d(v) - L(v).$$



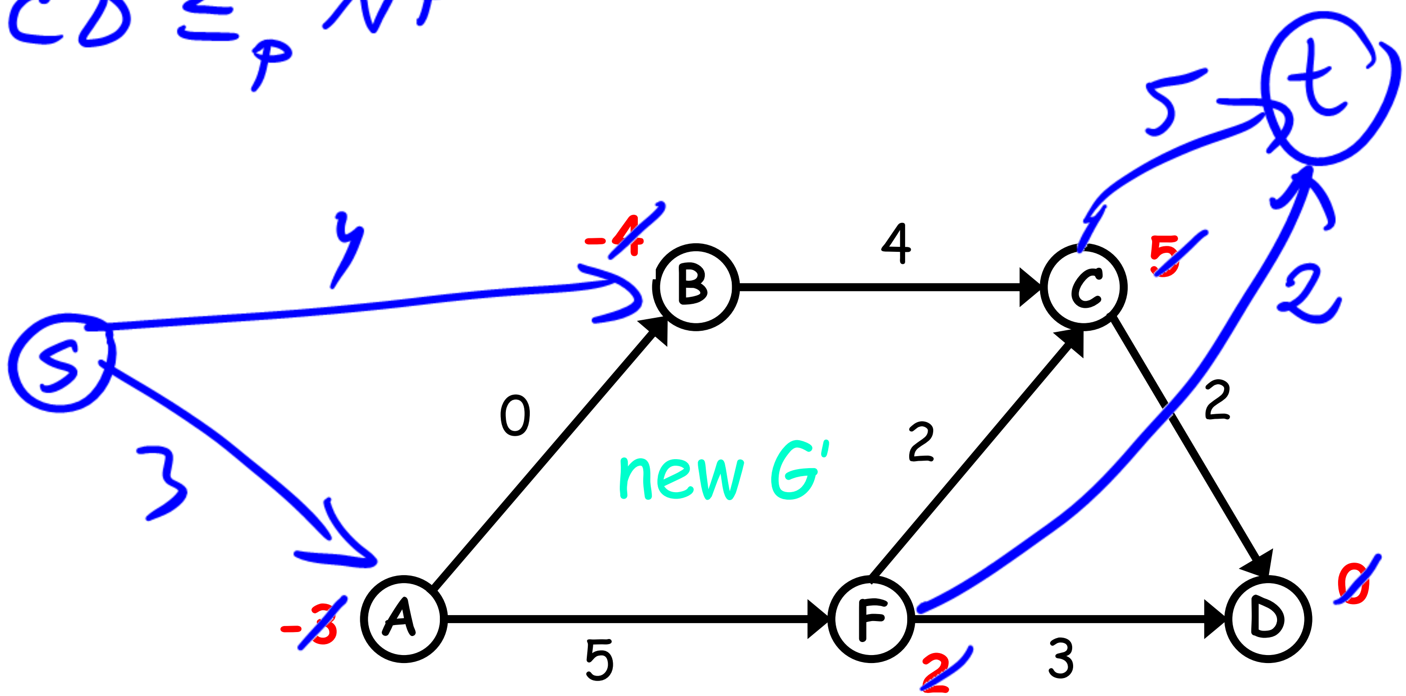
$$f^{in}(b) - f^{out}(b) = -5 = d(b)$$

$$d'(b) = -5 + (-2) + 3 = -4$$

Circulation with Demands and Lower Bounds

CDLB

$CDLB \leq_p CD \leq_p NF$



Claim: there is a feasible circulation in G iff there is a feasible circulation in a new graph G′.

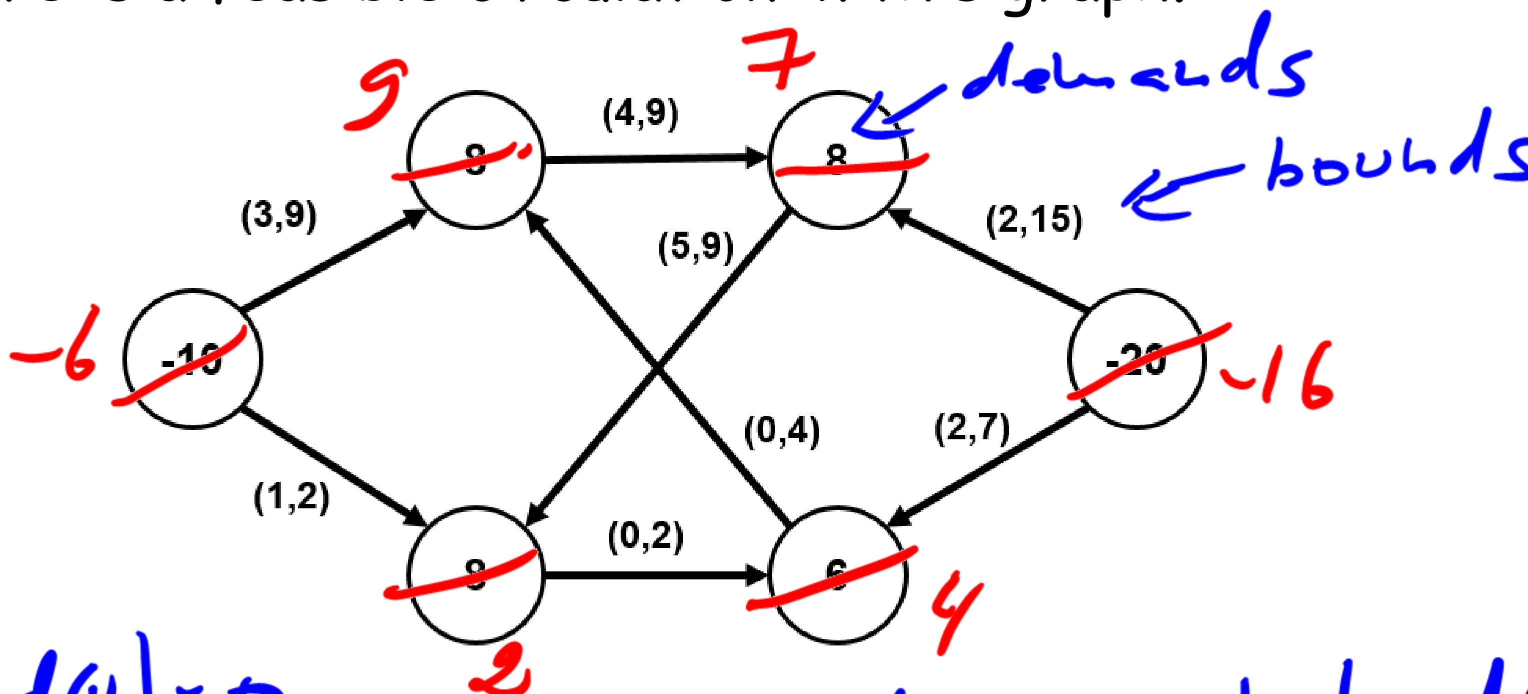
Circulation with Demands and Lower Bounds

Summary: given G with lower bounds, we:

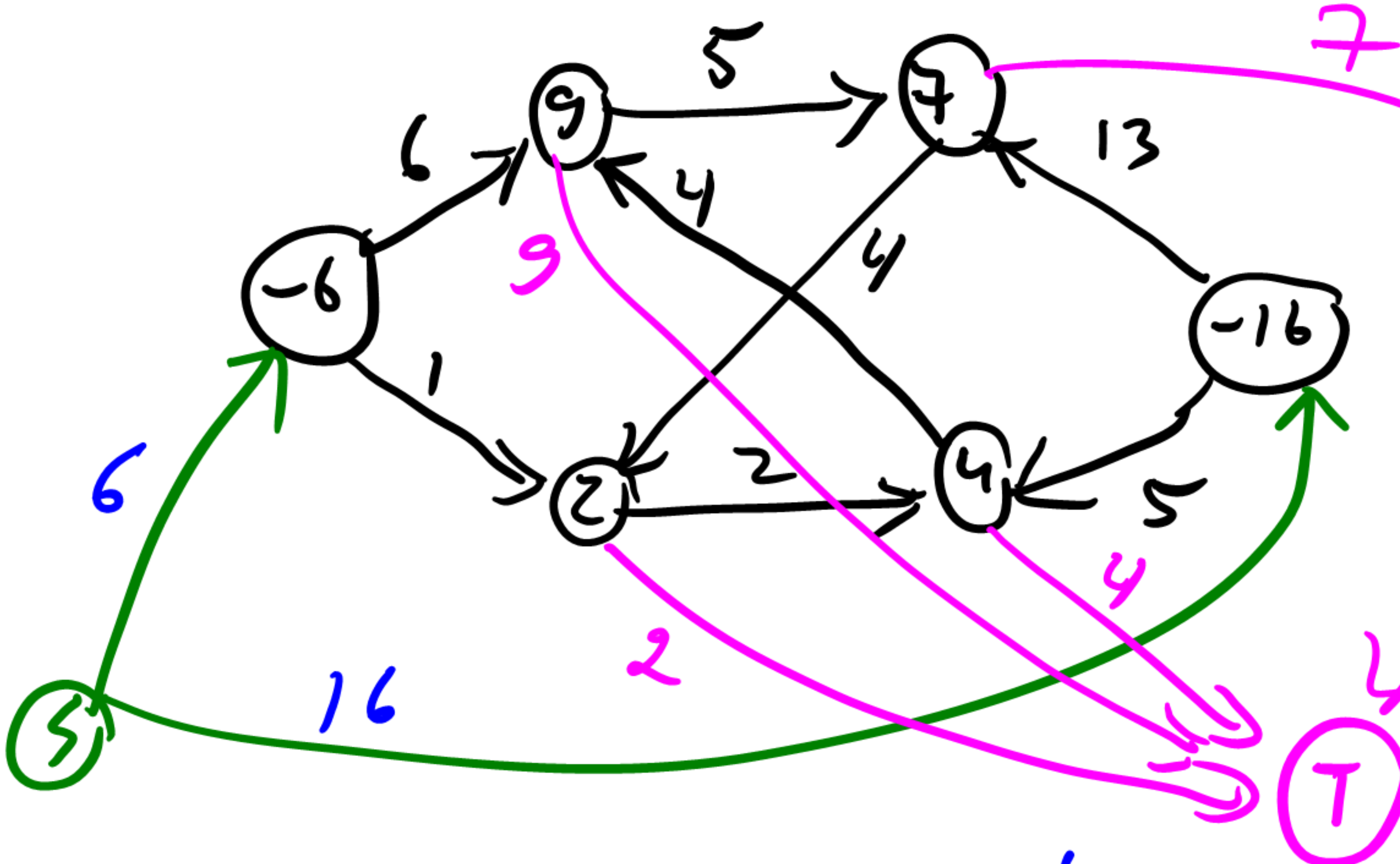
1. subtract lower bound $\ell(e)$ from the capacity of each edge
2. subtract $L(v)$ from the demand of each node
3. solve the circulation problem on this new graph to get a flow f.
4. add $\ell(e)$ to every $f(e)$ to get a flow for the original graph

Discussion Problem 4

Given the network below with the demand values on vertices and lower bounds on edge capacities, determine if there is a feasible circulation in this graph.



1. $\sum d(v) = 0$
2. remove lower bounds, recompute $d(v)$
3. max-flow

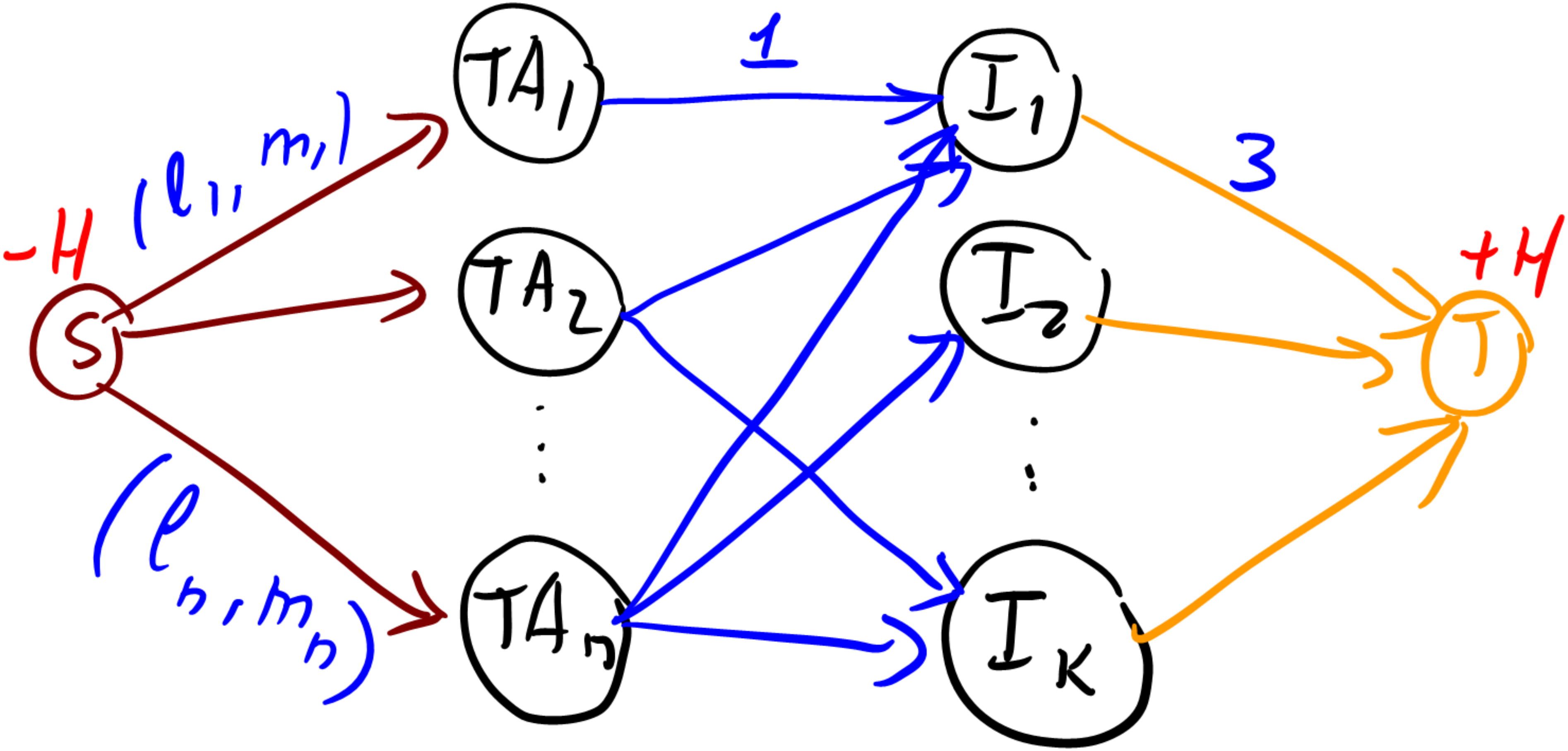


claim. $\exists$ circulation iff
max-flow $= 22$

Discussion Problem 5

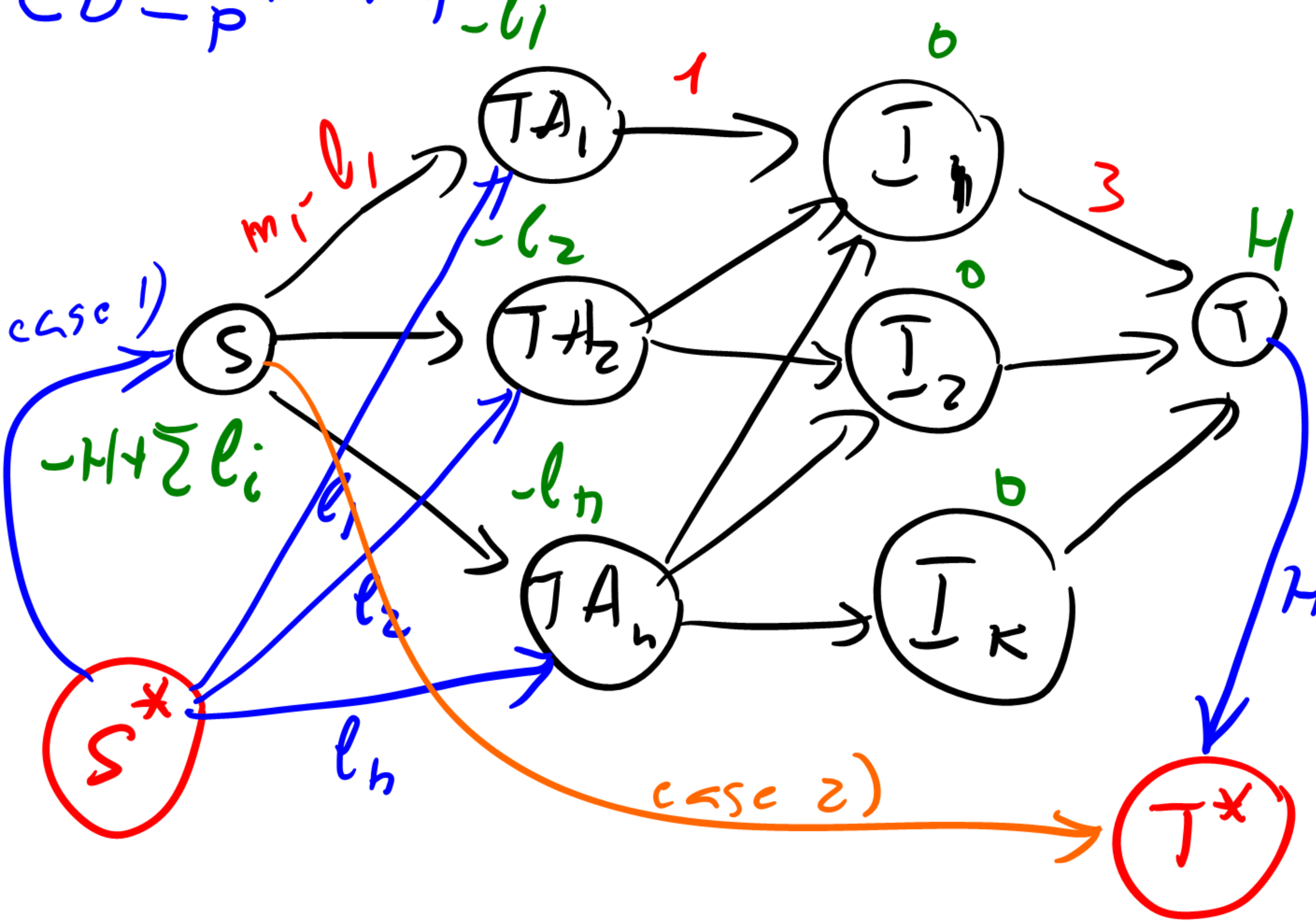
CSCI 570 is a large class with n TAs. Each week TAs must hold office hours in the TA office room. There is a set of k hour-long time intervals $I_1, I_2, \ldots I_k$ in which the office room is available. The room can accommodate up to 3 TAs at any time. Each TA provides a subset of the time intervals he or she can hold office hours with the minimum requirement of l_j hour 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.

$CDLB \leq_p CD \leq_p NF$



Does a circulatio exist?
Does a TA assignment exist?

$CD \leq_p$ max-flow



1) case $-H + \sum l_i < 0 \Rightarrow H > \sum l_i$

2) case $-H + \sum l_i > 0 \Rightarrow H < \sum l_i$

<u>Claim 1</u> Let $H \geq \sum l_i$.

TA assignment $\exists$ iff

the max-flow $= H$

<u>Claim 2</u> Let $H < \sum l_i$

TA assignment $\exists$ iff

the max-flow $= \sum l_i$

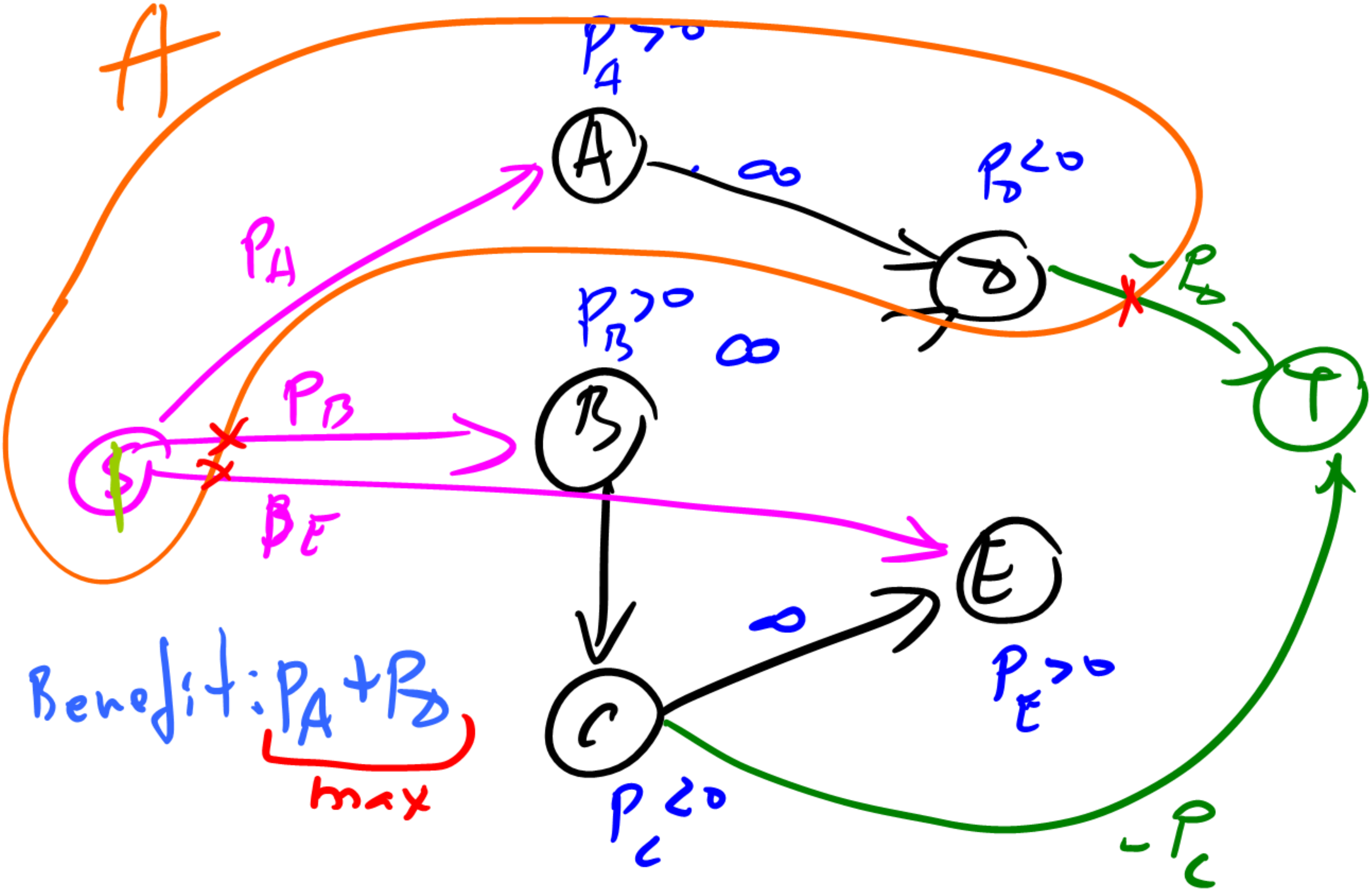
Discussion Problem 6

The computer science department course structure is represented as a directed acyclic graph $G = (V, E)$ where the vertices correspond to courses and a directed edge (u, v) exists if and only if the course u is a prerequisite of the course v. By taking a course w, you gain a benefit of p_w which could be a positive or negative number. Note, to take a course, you have to take all its prerequisites. Design an efficient algorithm that picks a subset $S \subset V$ of courses such that the total benefit is maximized.

benefit = $\sum p_w$, where $w \in S$.

$u \to v$

goal: max benefit



Benefit: $\underbrace{p_A + p_D}_{\text{max}}$

$$\mathrm{Cap}(A,B) = p_B + p_E - p_D \leftarrow \text{min}$$

How $C_{\text{ap}}(A,B)$ related to the benefit?

$$P_B + P_E - P_D = \underbrace{(P_A + P_B + P_E)}_{\text{all positive classes}} \underbrace{- P_A - P_A}_{\text{min benefit}}$$

NF

$$\underline{\min}(\text{cut capacity}) =$$

$$= \min\left(\sum_{P_K>0} P_K - \text{benefit}\right) =$$

$$= \sum_{P_K>0} P_K + \min(-\text{benefit}) = \sum_{P_K>0} P_K + \text{max benefit}$$

$$\max(\text{benefit}) = \underbrace{\text{min-cut}}_{\substack{FF \\ \text{max-flow}}} - \sum_{p_K > 0} p_K$$