Analysis of Algorithms

V. Adamchik CSCI 570 Spring 2020

Lecture 10 University of Southern California

Network Flow - 2

Reading: chapter 7

Exam Statistics

Number of submitted grades: 570 / 582

Minimum: 24 %

Maximum: 94 %

Average: 74.28 %

Median:76 %

Standard Deviation:9.75 %

C-	C	C+	B-	B	B+	A-
54	55	64	69	74 - 79	84	89

The Ford-Fulkerson Algorithm

$c \in \mathbb{N}$

$f \in \mathbb{N}$

Algorithm. 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

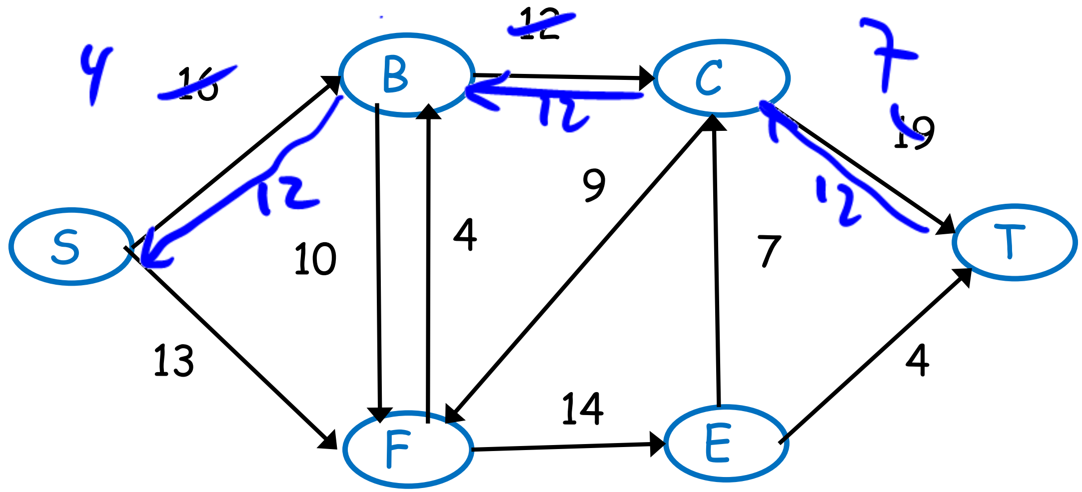
Def. Flow

$f : E \to \boxed{\mathbb{R}^+}$

1)

2)

G_f



s-B-C-T

$c_f = c - f$

$c(BC) = 12 - 12 = 0$

Cuts and Cut Capacity

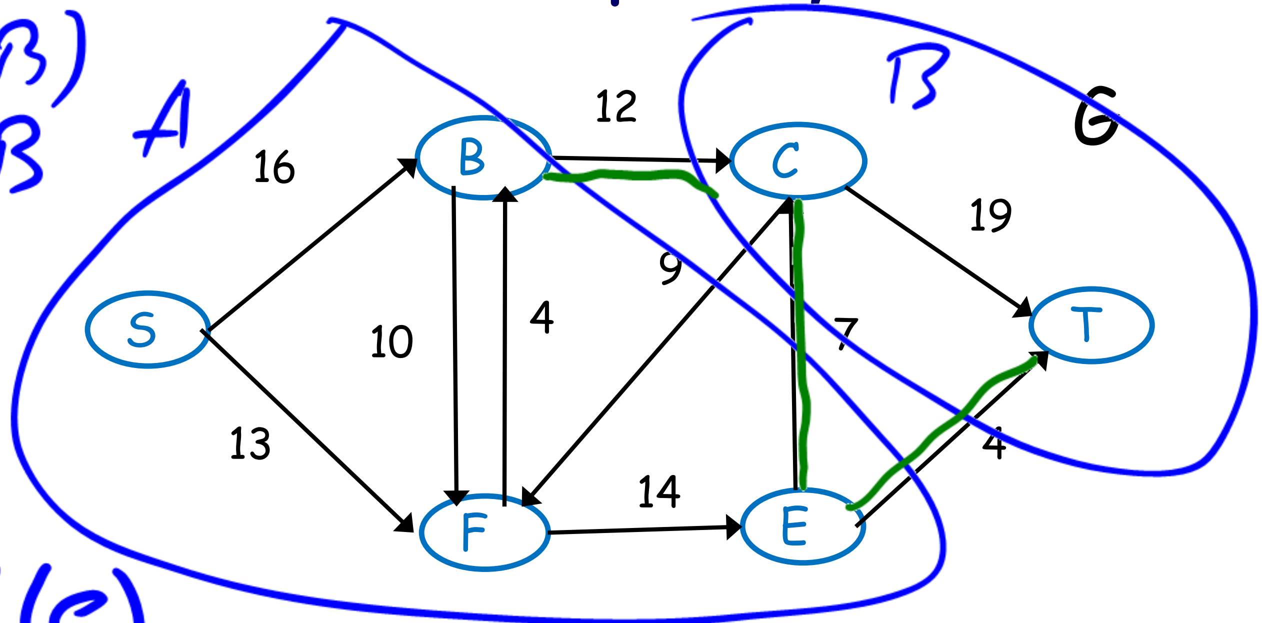
Def. Cut(A,B)
$s \in A,\ t \in B$

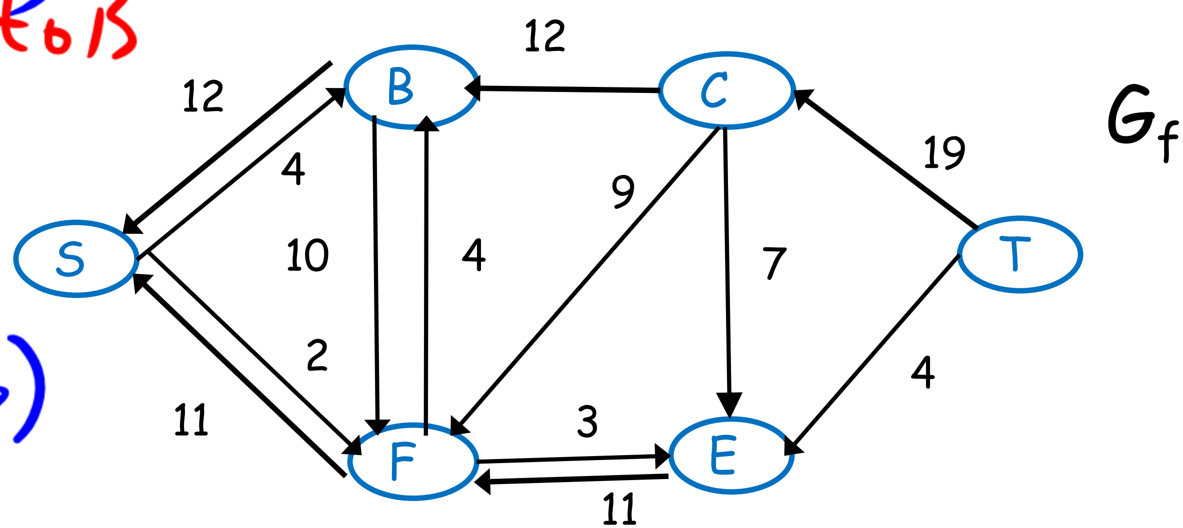
Def. Cut Cap

$Cap(A,B) = \sum c(e)$ → e from A to B

$= 12+7+4 = 23$

Lemma. $\forall$ flow, cut
$|f| \le cap(A,B)$





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:

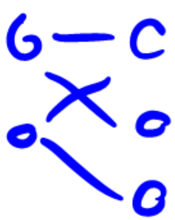
& space

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

BM $\leq_p$ NF

G'

G



$\Longleftrightarrow$

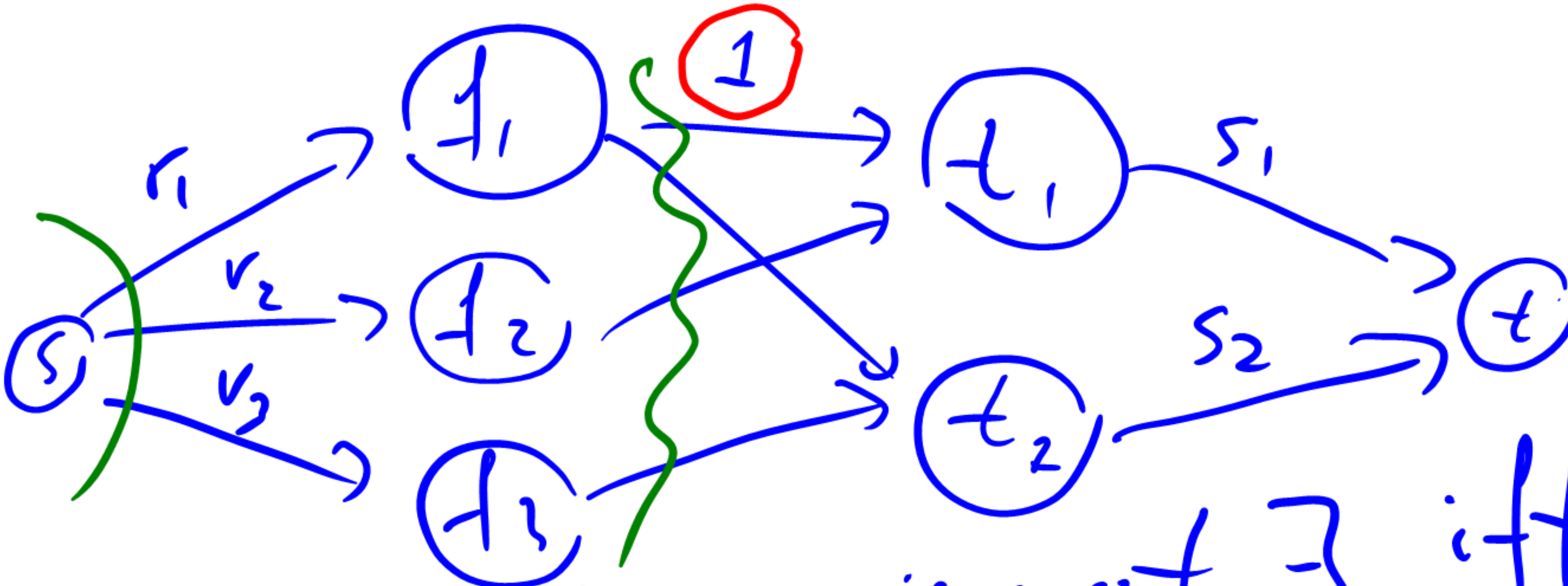
Solving by reduction to NF

s, t, c, V, E

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 $f_1, f_2, \ldots, f_n$ and m tables $t_1, t_2, \ldots, t_m$. The i-th family f_i has r_i relatives and the j-th table t_j has s_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?



Claim. A seating assignment $\exists$ iff the max-flow $= \sum_{i=1}^{n} r_i$

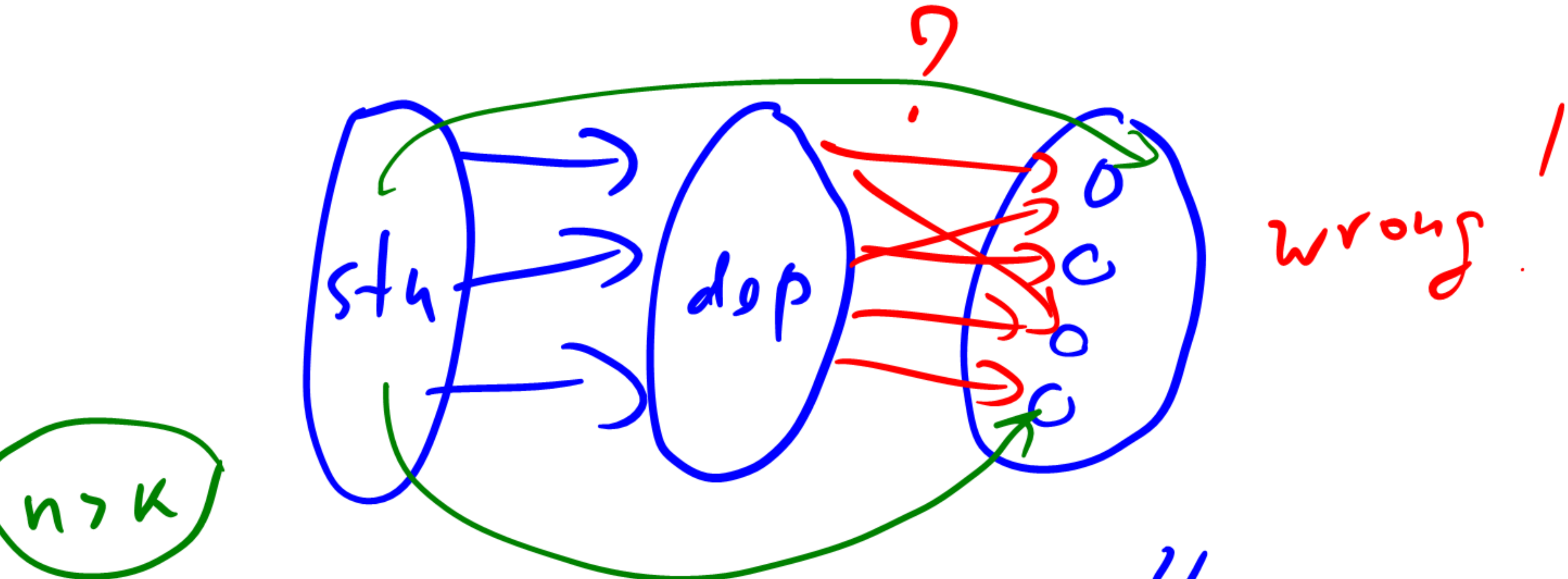
Proof.

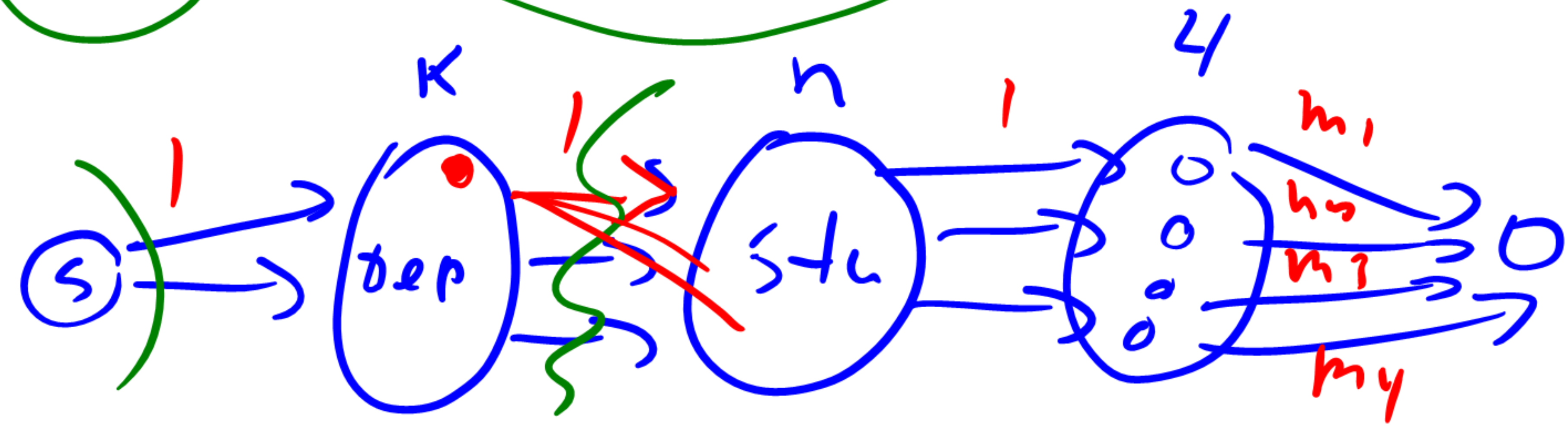
$\Rightarrow$ Given a seating assignment,

$\Leftarrow$ Given the max-flow.

Discussion Problem 2

There are n students in a class. We want to choose a subset of k students as a committee. There has to be m_1 number of freshmen, m_2 number of sophomores, m_3 number of juniors, and m_4 number of seniors in the committee. Each student is from one of k departments, where $k = m_1 + m_2 + m_3 + m_4$. 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.





$K = m_1 + m_2 + m_3 + m_4$

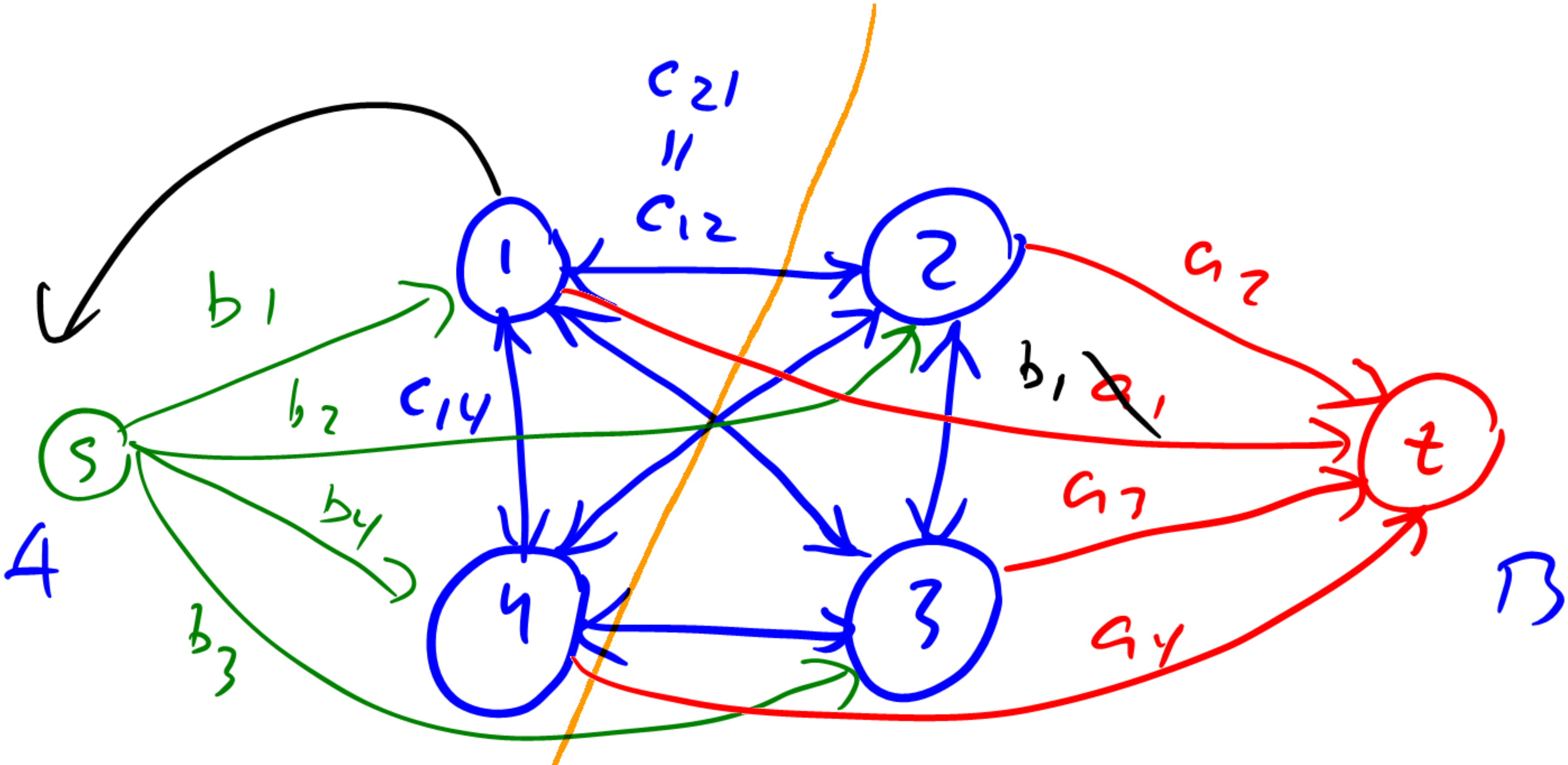
Runtime: $O(|f| \cdot E) = O(K \cdot n)$

#1 HW 4

Pr. 3

Discussion Problem 3

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, $\boxed{c_{ij} > 0}$, 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.



min

$\uparrow$

Ky

$$\mathrm{Cap}(A,B) = a_1 + a_4 + b_2 + b_3 + c_{12} + c_{13} + c_{24} + c_{43}$$

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

Proof.

$\Rightarrow$ by construction

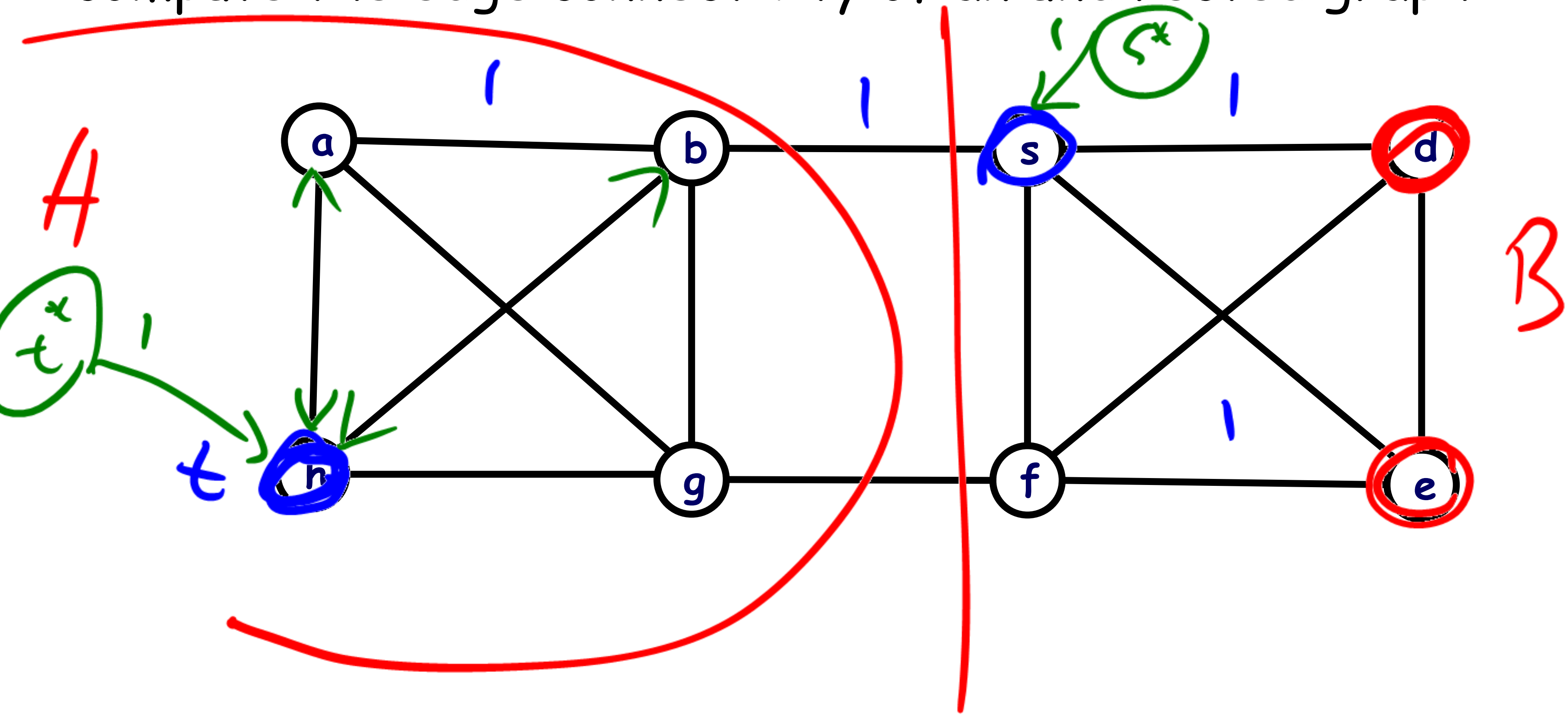
$\Leftarrow$

Runtime: Assuming FF

$O(|f| \cdot E) = O(n^2 \cdot |f|)$

Discussion Problem 4

The edge connectivity of an undirected graph G = (V, E) is the *minimum* number of edges whose removal disconnects the graph. Describe an algorithm to compute the edge connectivity of an undirected graph



Algorithm. Fix $\forall s \in V$

~~For $\forall x \in V$ (as a source)~~

For $\forall y \in V$ (as a target)

Run FF

Count min-cut edges

$\mathrm{Min}\{mc_1, mc_2, \dots, mc_{v^2}\}$

Global min-cut

Runtime: Assuming FF

$O(|f| \cdot E) = O(V \cdot E)$

Total: $O(V^{\cancel{2}} \cdot VE)$

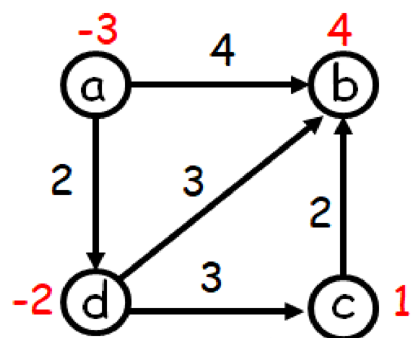
Circulation

Suppose that there can be a set S of sources generating flow, and a set T of sinks that can absorb flow. As before, there is an integer capacity on each edge.

We call this *circulation* since we have s-t flow as well as t-s flow.
Goal is to find a circulation.

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 $d(v) < 0$.



$d(a) = -3$

$d(b) = 4$

Def. A circulation with demands $f: E \to R^+$

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

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

Necessary Condition

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

Demand = Supply

Proof.

Conservation:

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

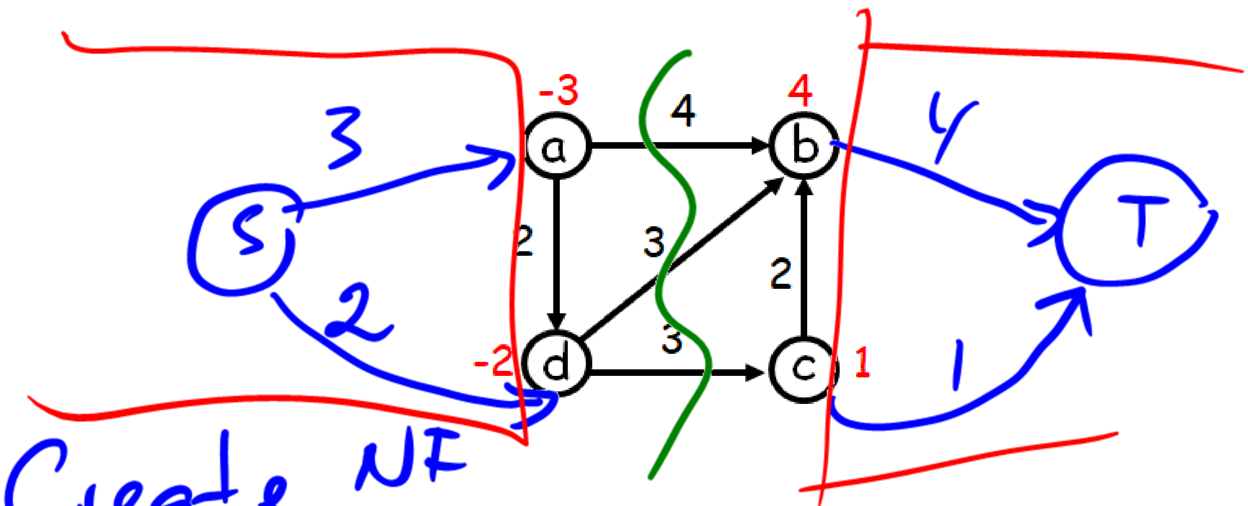
$$\sum_{v \in V} \left[f^{in}(v) - f^{out}(v) \right] = \sum_{v \in V} d(v)$$

$$0 =$$

$$\to x \to y \to$$

$$f^{in}(x) - \cancel{f^{out}(x)} + \cancel{f^{in}(y)} - f^{out}(y)$$

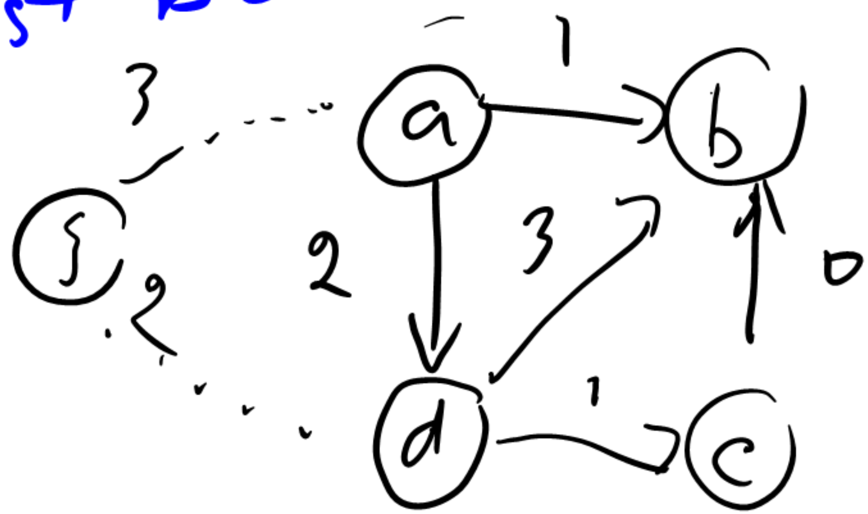
Reduction to Flow Problem



1. Create NF
2. Run FF

max-flow must be 5

3. Claim.



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.

$\Rightarrow$

$\Leftarrow$ max-flow $\to$ circulation

Run FF

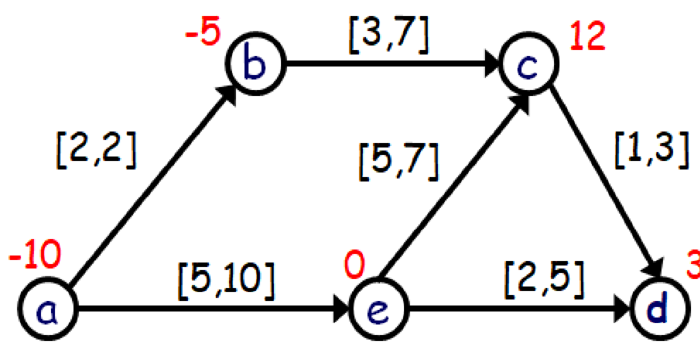
Remove blue edges, incident to S, T

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.

G has a circulation iff max-flow = 7

G

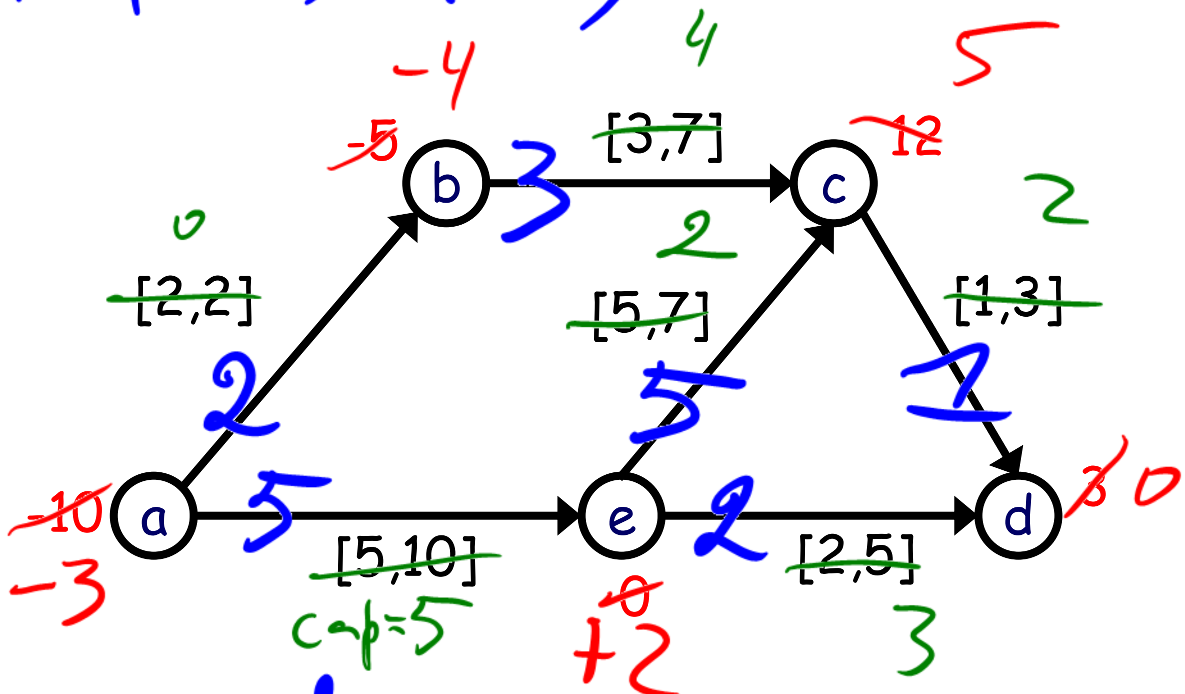


capacity law: $\ell(e) \le f(e) \le c(e)$

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

Circulation with Demands and Lower Bounds

1. Push $f(e) = \ell(e)$



2. change demands.

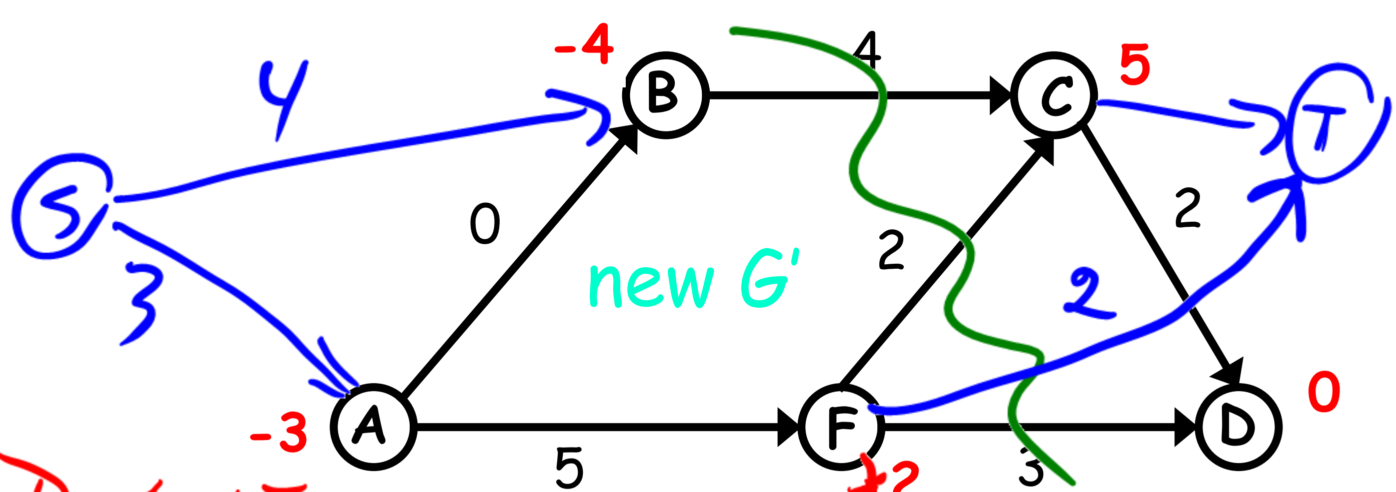
Circulation with Demands and Lower Bounds

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

$\boxed{d'(v) = d(v) - L(v).}$

G''

$\longrightarrow p$



$CDLB \leq_p CD \leq_p NF$

<u>Claim:</u> 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:

subtract lower bound $\ell(e)$ from the capacity of each edge

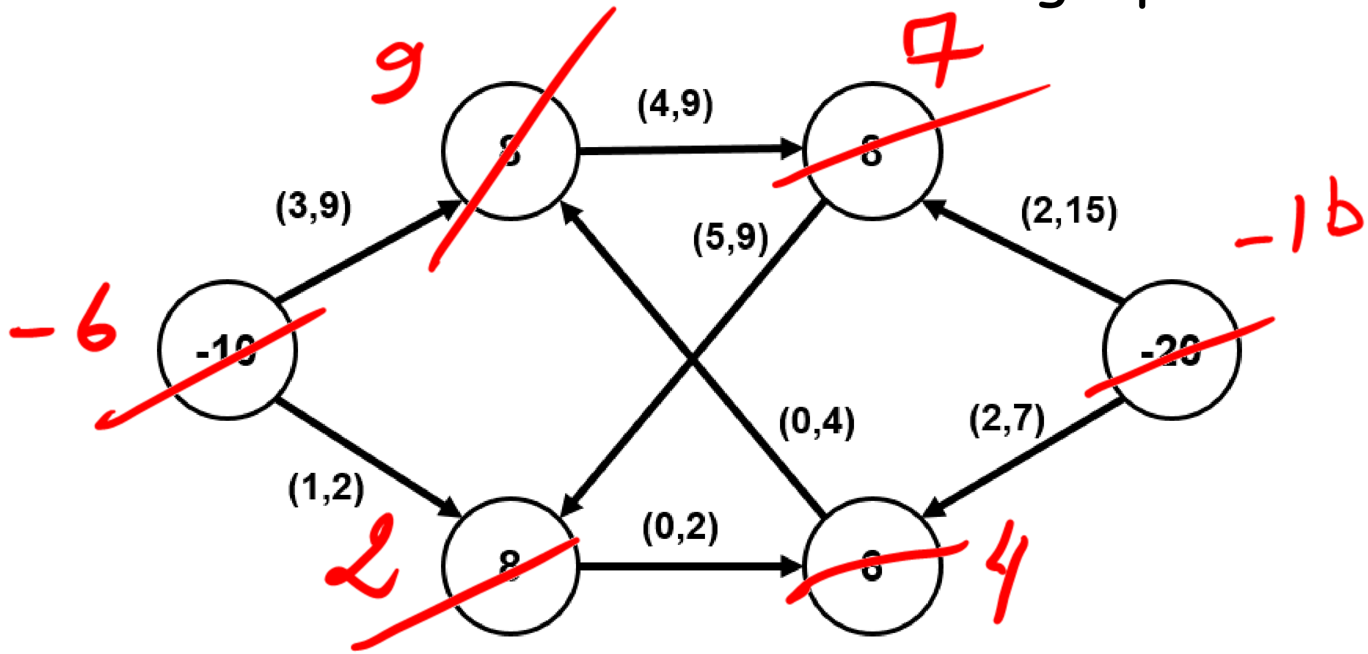
subtract $L(v)$ from the demand of each node

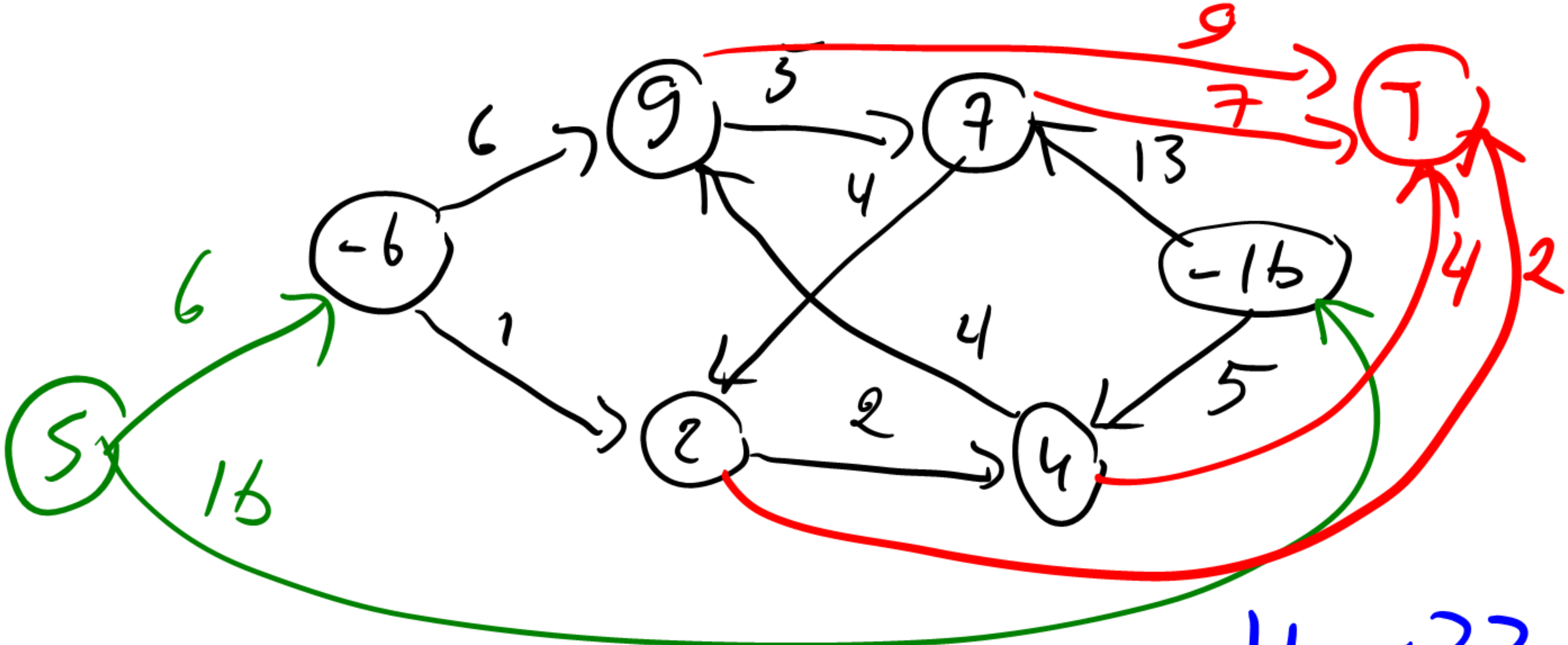
solve the circulation problem on this new graph to get a flow f.

add $\ell(e)$ to every $f(e)$ to get a flow for the original graph

Discussion Problem 5

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.





<u>Claim</u>. $\exists$ circulation iff max-flow = 22

Yes~! verify!!!

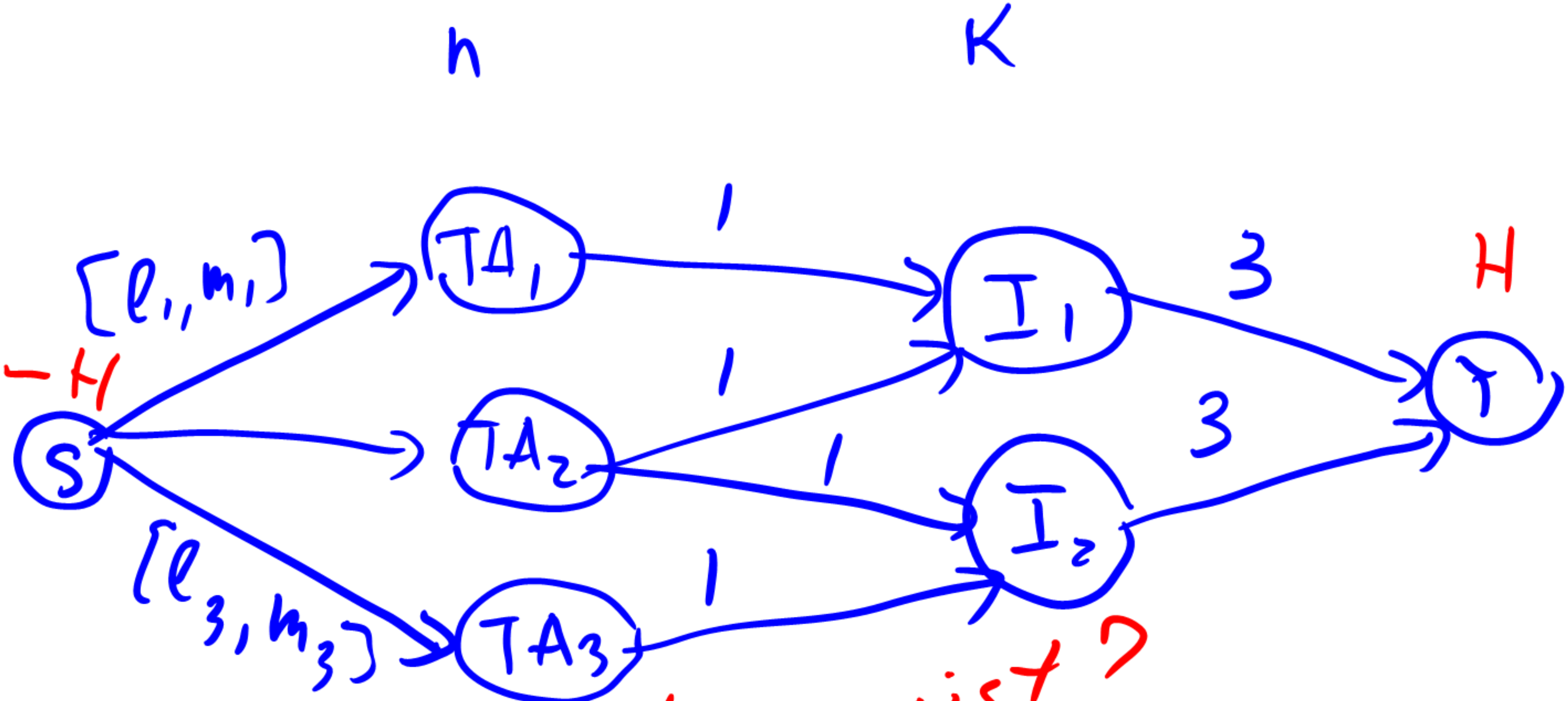
$\boxed{T/F}$ A circulation is a flow.

$d(v) = 0$
$\ell(v) = 0$

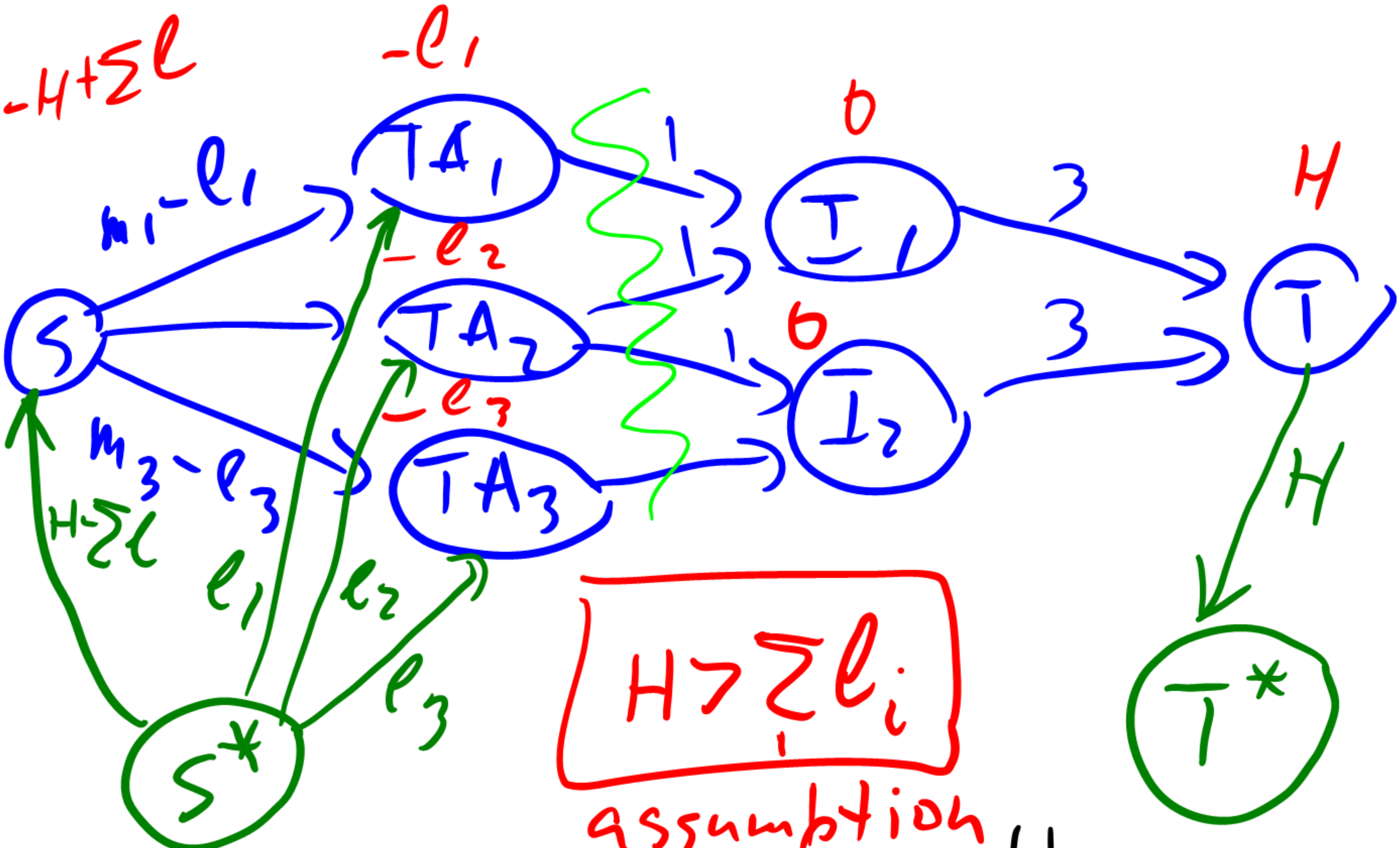
$\boxed{F}$

Discussion Problem 6

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.



Is a circulation exist?



Claim: TA assignment $\exists$ iff max-flow $= H$

Case 2. $H < \sum l_i$

vertex s is connected to T^*

Claim. $\exists$ TA assignment iff

max-flow $= \sum l_i$

How do you find assignment?

Can we solve circulation problem in polynomial time?

Yes.