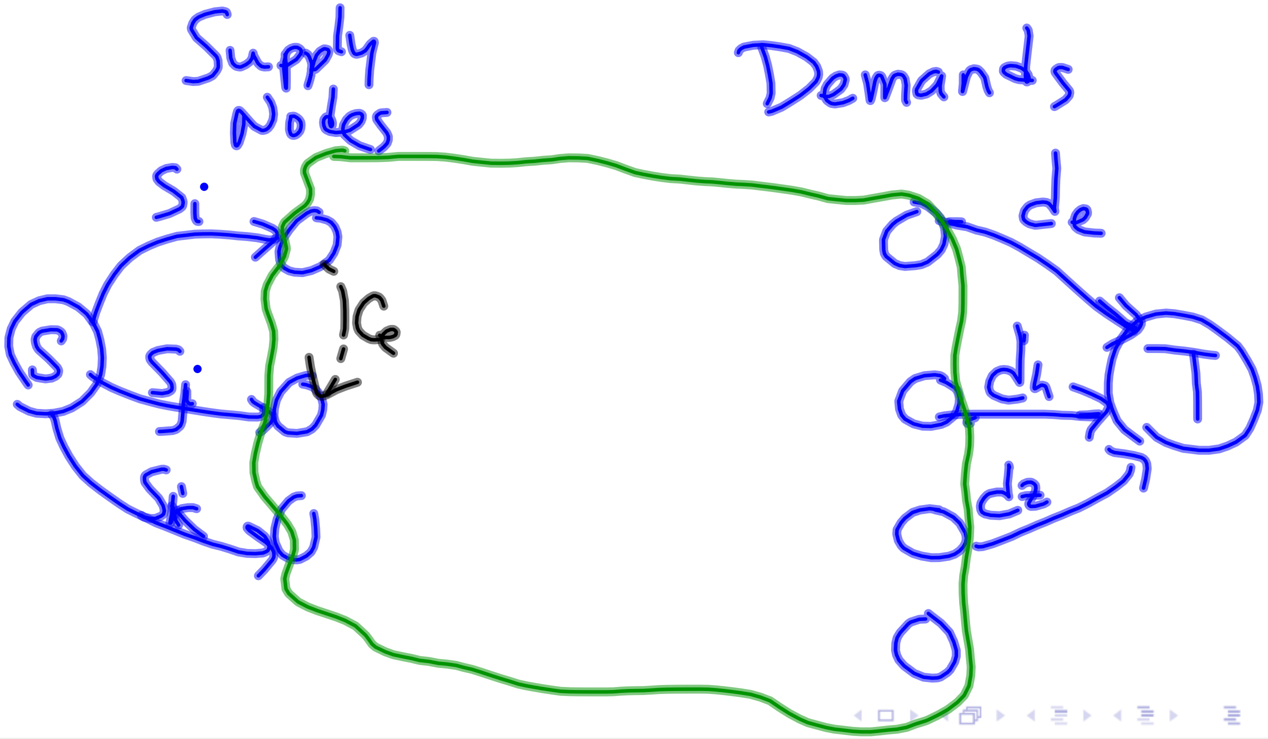
Warm-Up

Try questions 1 & 2 in section "Supply, Demand, and Lower Bounds" of Network Flow packet.

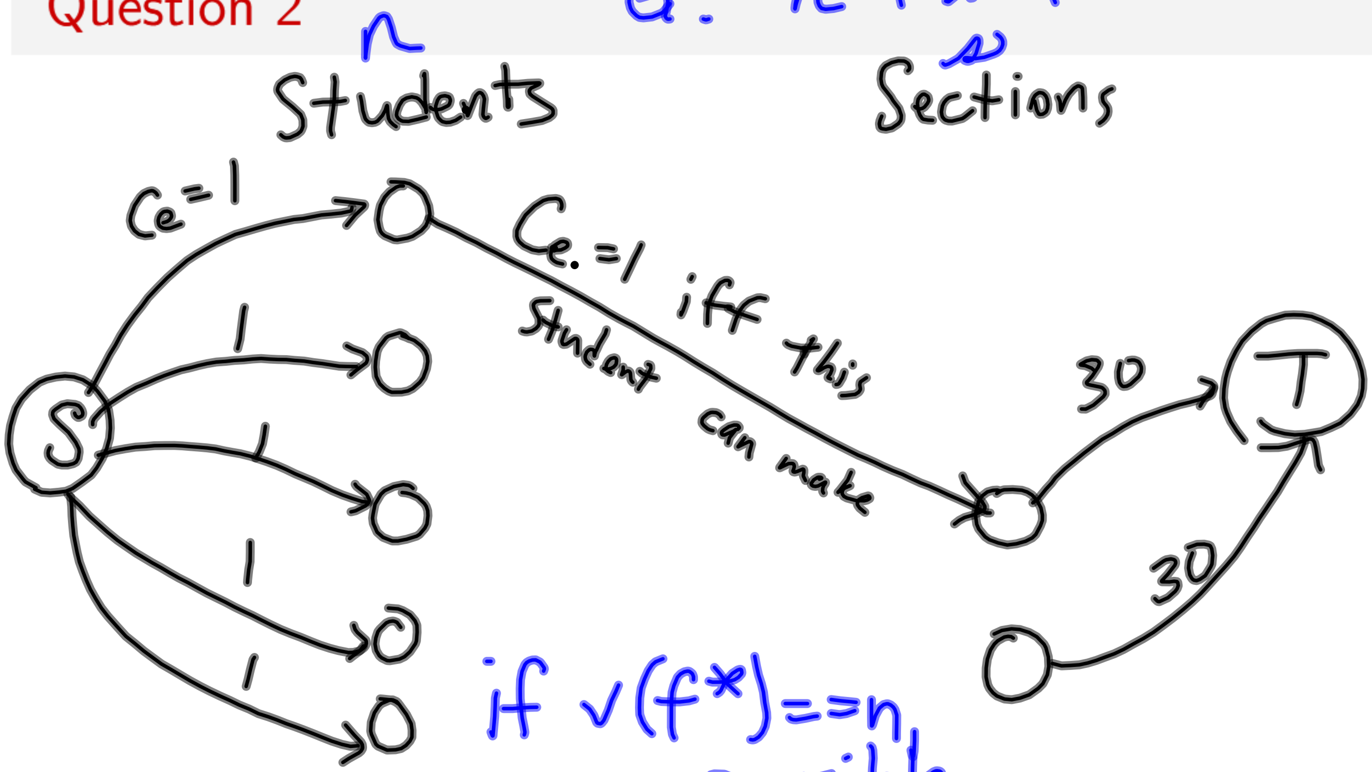
- max flow / min-cut
- edge disjoint paths
- bipartite matching
- supply/demand

Question 1 $(G, \{c_e\}, \{s_i\}, \{d_i\})$



Question 2

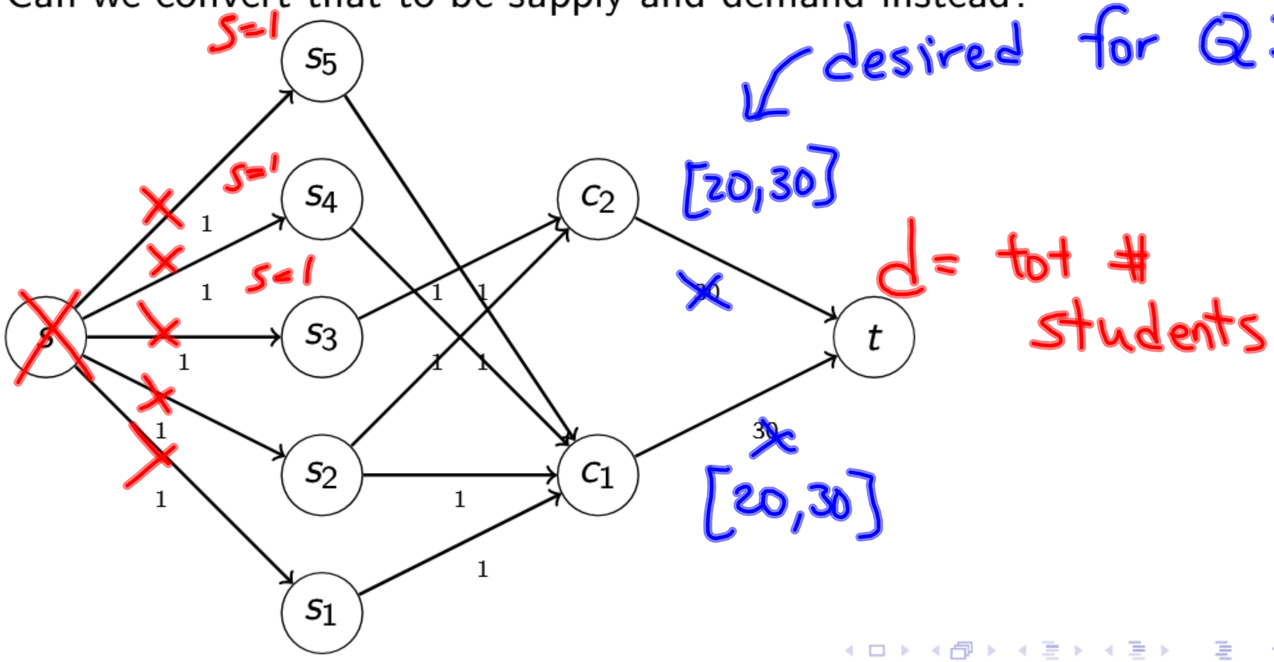
G: n + s + 2 vert.



if $v(f^*) == n$
possible

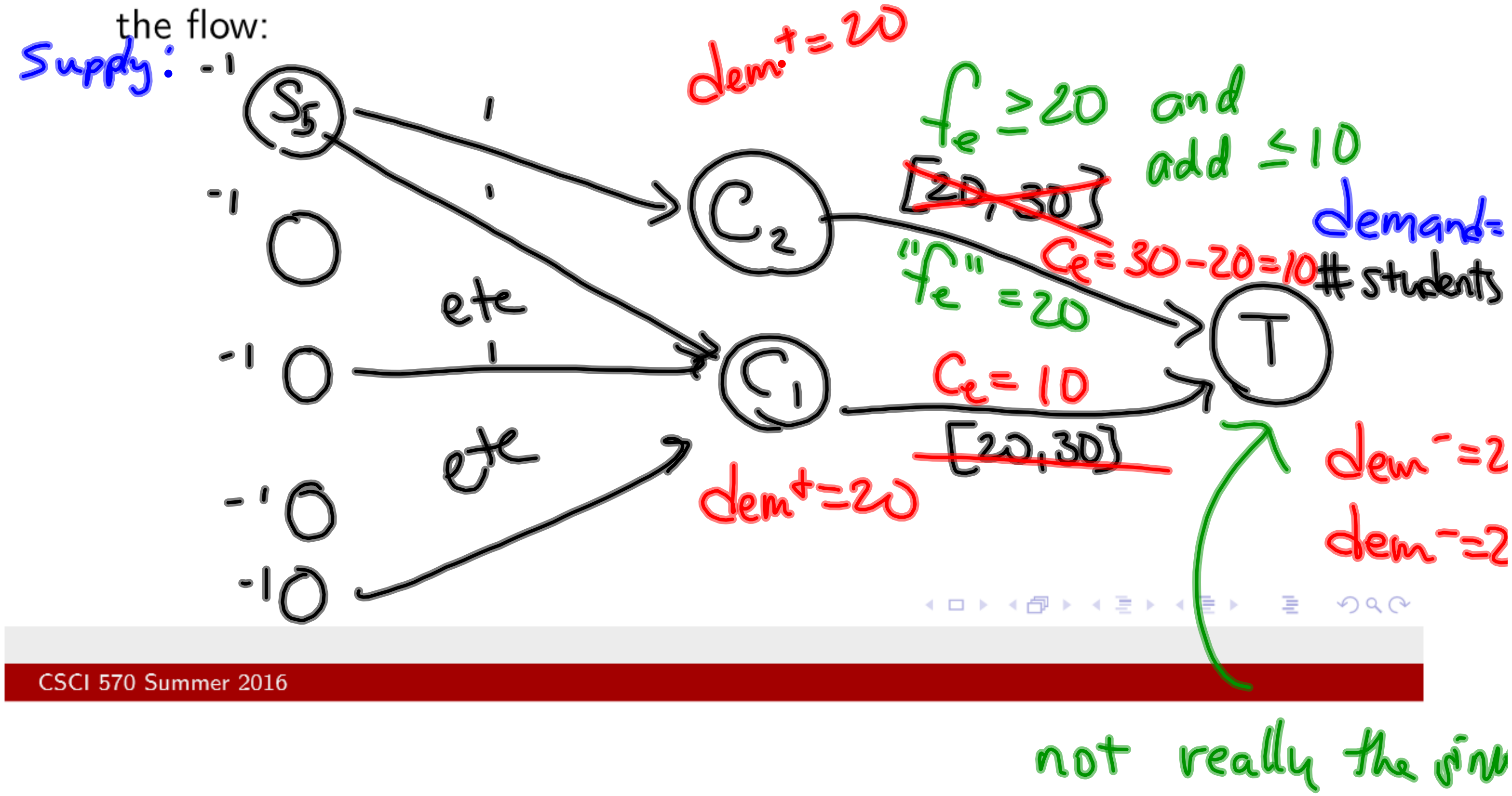
Question 2

Can we convert that to be supply and demand instead?



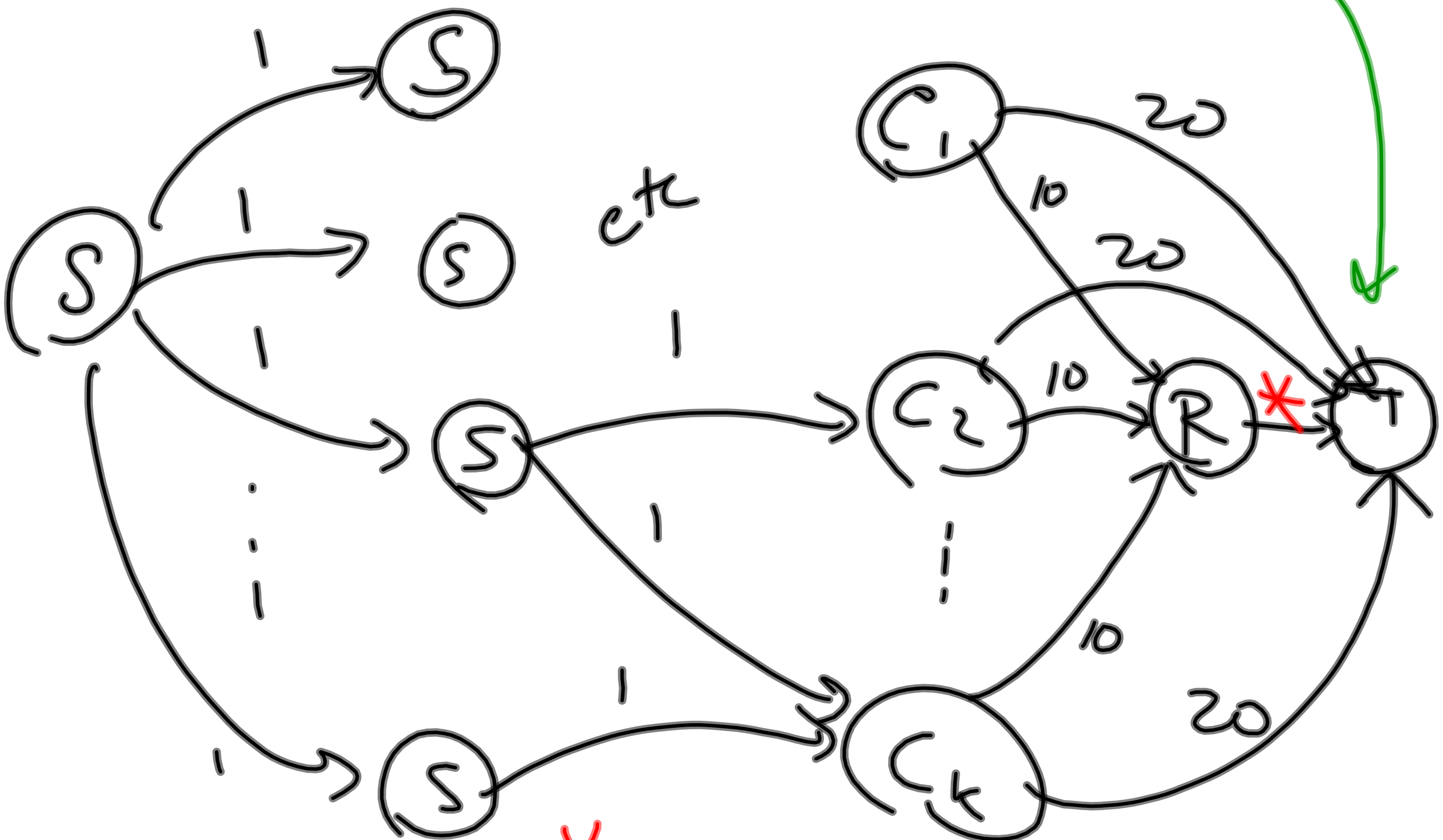
Question 3

Let's amend our solution from Question 2 to put a *lower bound* on the flow:



finally reduced graph

really the sink!



$* \; C_e = \#\text{students} - 20 \cdot \#\text{classes}$

Baseball Elimination Problem

Team i	Wins w_i	Games Left g_i	Schedule ($g_{i,j}$) LA	Oak	Sea	Tex
Los Angeles	81	8		1	6	1
Oakland	77	4	1		0	3
Seattle	76	7	6	0		1
Texas	74	5	1	3	1	

Can Texas team finish in first place?

No. max 79 (= 74 + 5) wins.

Baseball Elimination Problem

Team i	Wins w_i	Games Left g_i	Schedule ($g_{i,j}$) LA	Oak	Sea	Tex
Los Angeles	81	8		1	6	1
Oakland	77	4	1		0	3
Seattle	76	7	6	0		1
Texas	74	5	1	3	1	

Can Oakland team finish in first place?

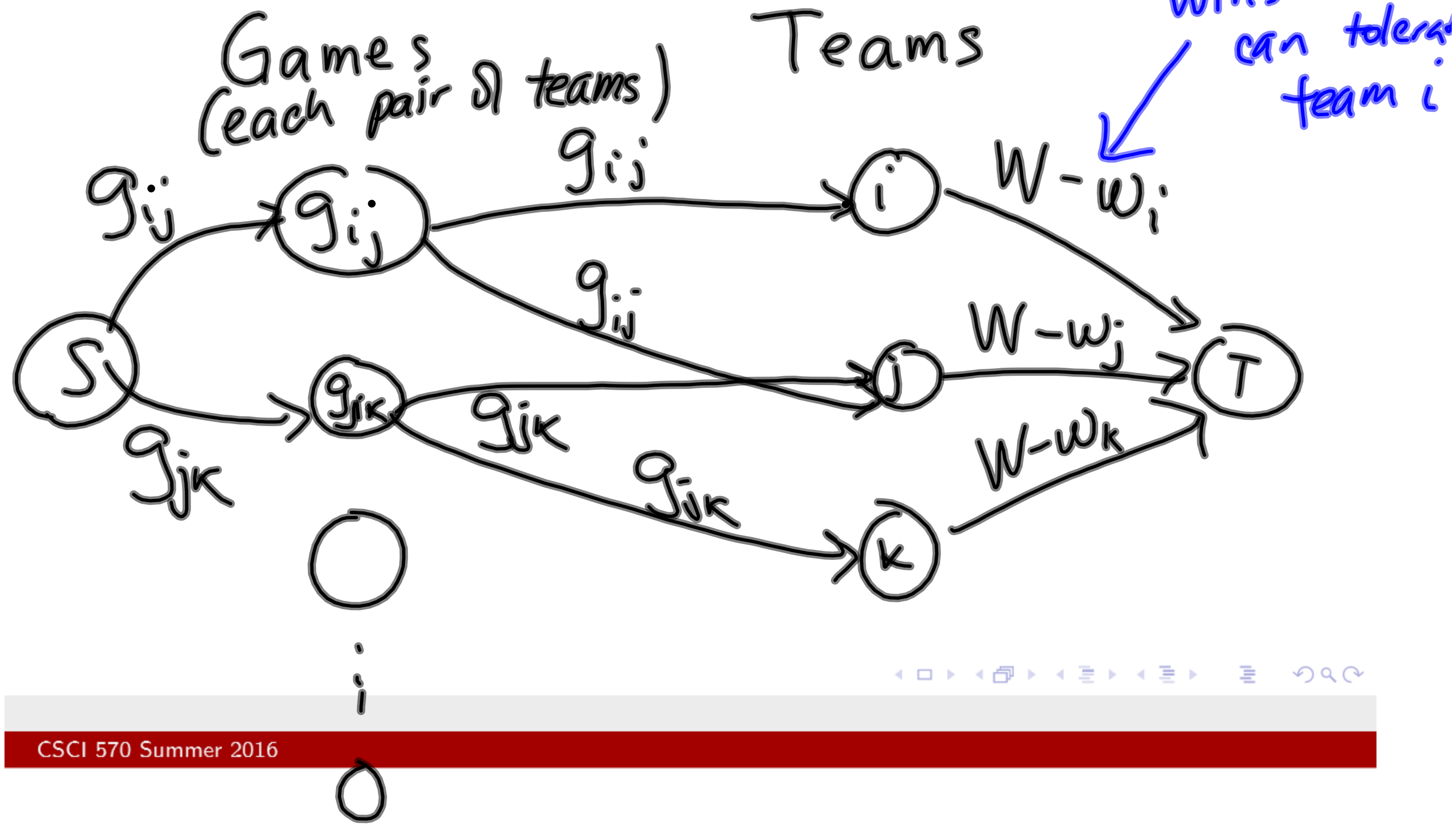
L.A. + SEA tot wins = 81 + 76 + 6 = 163
so at least one of them has 82 wins

Has team k been eliminated?

How could I convince you that a team *does* have a chance to finish in first place?

map games to winners
so "we're" 1st

W = wins if we "win out"

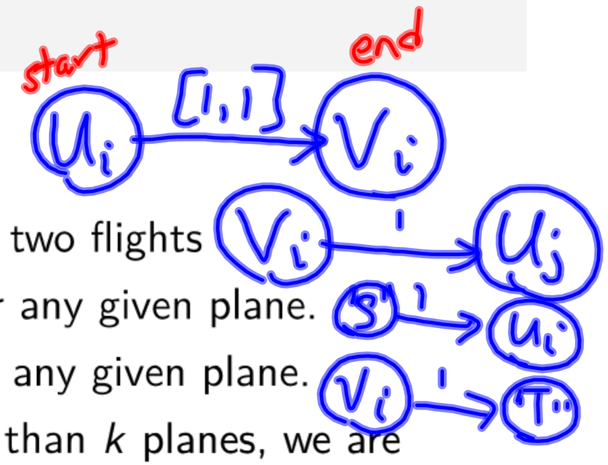


Example

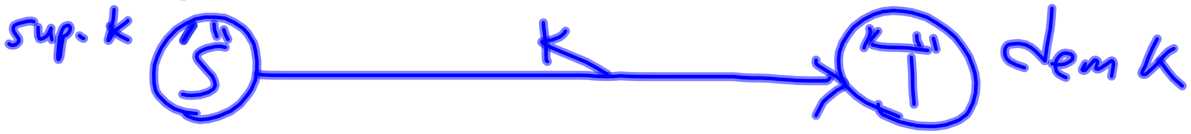
- Boston (6AM) to Washington D.C. (7AM)
- Philadelphia (7AM) to Pittsburgh (8AM)
- Washington D.C. (8AM) to Los Angeles (11AM)
- Philadelphia (11AM) to San Francisco (2PM)
- San Francisco (2:15 PM) to Seattle (3:15PM)
- Las Vegas (5PM) to Seattle (6PM)

We can do this with $k = 4$ airplanes. If we have more, that's also okay.

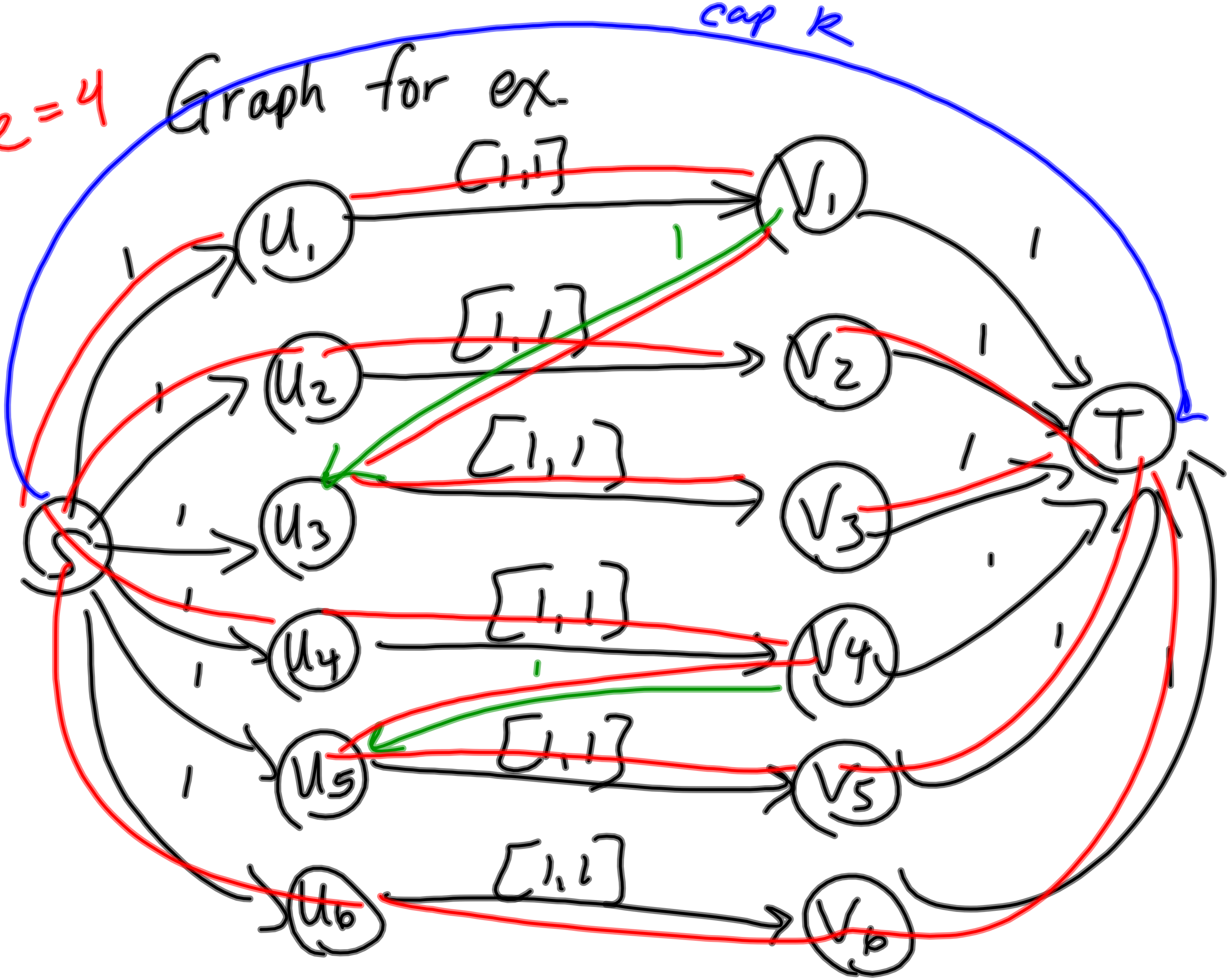
Ideas to represent in the graph



- Each flight on the list must be served
- The same plane can sometimes perform two flights
- Any flight can be the first of the day for any given plane.
- Any flight can be the last of the day for any given plane.
- If we can service every flight with fewer than k planes, we are under no obligation to use all k.

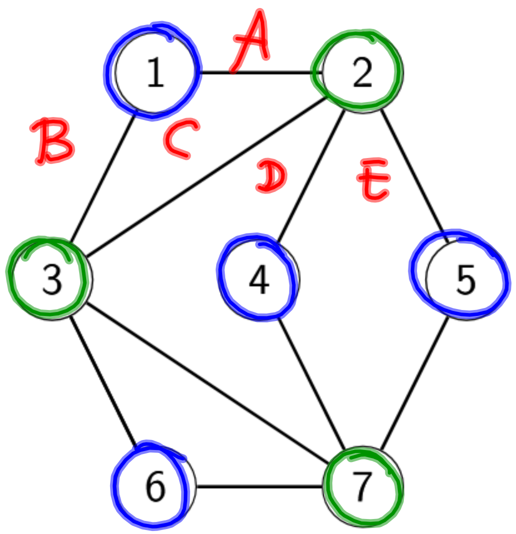


$k = 4$ Graph for ex.



Independent Set and Vertex Cover

Find an independent set of size 4 and a vertex cover of size 3 in this graph:



Set Cover instance

Can you confirm a solution?

- Suppose I claim G has an INDEPENDENT SET of size k. What evidence should I provide of my claim?
  Give a subset of vertices
- Could you write an algorithm to verify such a claim?
  Input: G, k, and the evidence from the first point.
  Output: True or false, indiciating if the evidence really confirms an independent set of size k.

Verifier for INDEPENDENT SET

Certificate: V', a set of vertices.

Verifier:

if $V' \not\subseteq V$ **then** $\Big]$ $O(|V|)$? $O(n^2)$?
  return false
if $|V'| \neq k$ **then** $O(1)$
  return false
for all edges $e = (u, v) \in E$ **do** $\}$ $O(m)$
  if $u \in V'$ and $v \in V'$ **then** $\}$ $O(1)$? $O(k)$? $O(n)$ $\}$ $O(nm)$
    return false
return true

Which is easier?

Which do you think is easier to write an algorithm for?

Vertex Cover(G, k) Independent Set(G, k)

return IndSet(G, |V|-k) You?

return vc(G, |V|-k);

Which is easier?

Which do you think is easier to write an algorithm for?

$\textsc{Vertex Cover}(G, k)$	$\textsc{Independent Set}(G, k)$
return $\textsc{Ind. Set}(G, n-k)$	return $\textsc{Ver. Cov}(G, n-k)$

Coincidence?

Claim: a graph G has an INDEPENDENT SET of size k if and only if it also has a VERTEX COVER of size $n - k$.

Part 1 of proof: If a graph G has an INDEPENDENT SET of size k, it also has a VERTEX COVER of size $n - k$.

Let $\mathcal{I}$ be I.S. size k.

Let $C = V - \mathcal{I}$ size $n-k$

$\forall e=(u,v)$ $u \notin \mathcal{I}$ or $v \notin \mathcal{I}$ or both

$\longrightarrow$ $u \in C$ or $v \in C$ or both

$\therefore$ C is V.C. of size $n-k$

Coincidence?

Claim: a graph G has an INDEPENDENT SET of size k if and only if it also has a VERTEX COVER of size $n - k$.

Part 2 of proof: If a graph G has VERTEX COVER of size $n - k$, it also has a an INDEPENDENT SET of size k.

(mirror prev slide)

Set Cover

Can you select three of the following sets in such a way that each letter from 'A' through 'J' is in at least one chosen set?

Set Number	Elements
1	A B
2	A C D E
3	B C F I
4	D G
5	E H
6	I J
7	F G H J

Is Set Cover easier or harder than Vertex Cover?

Can we use SET COVER to solve VERTEX COVER?

see ex on a few slides ago

Imagine I give you a solution to SET COVER and solve:

VERTEX COVER(G, k)

for each vertex v
  create set v = {}
for each edge (u,v)
  create element (distinct)
  add elt to sets u,v
Call set cover(sets, K)

3-Sat

$$\phi = (x_1 \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee x_3)(x_1 \vee x_2 \vee x_3)$$

Variable	Truth Value
x_1	True or False
x_2	True or False
x_3	True or False
~~x_4~~	~~True or False~~