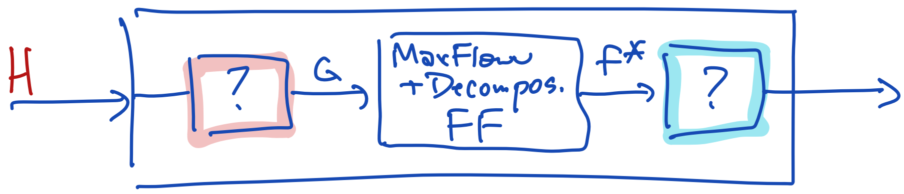
HW8 out tonight → due next wed

Nov 11 drop deadline for grads

Applications of maxflows/mincuts

- Edge-disjoint paths
- Vertex " "
- Bipartite matching

Reduction



- Transform input (algo)
- Transform output (algo)
- Time analysis in terms of original input
- Proof of correctness (we don't need this in HW, but you do)

Features in solutions ⇄ paths in max flow

Exam Scheduling

Input:

n	classes
r	rooms
t	time slot
p	proctor

E[1..n] enrollment

S[1..r] # seats

A[1..t, 1..p] = availability

A[k, l] = True means proctor l is available at time k

- Every class needs to be scheduled
- E ≤ S for each exam
- ≤ 1 exam per room per time slot
- Each proctor can watch ≤ 5 exams, only when available

Output: room, time, proctor for every class.

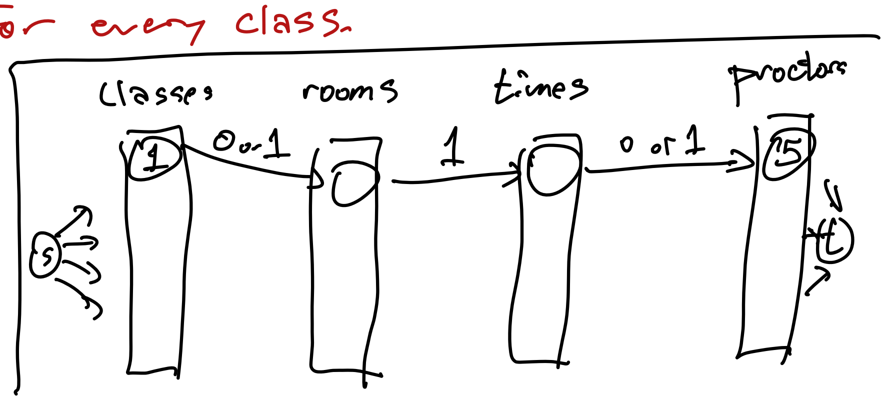
Set of tuples (i, j, k, l)

i – class one per exam

j – room

k – time

l – proctor



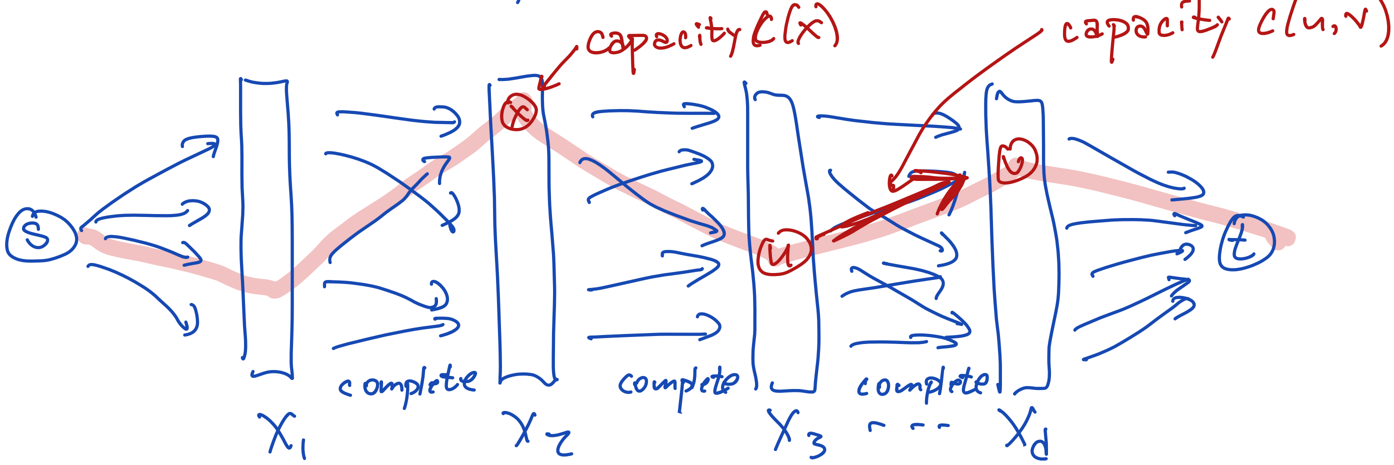
Tuple Selection

Finite sets $X_1\ X_2 \dots X_d$ resources

- $c(x)$ for each $x \in X_i$ for all i ← capacity
- $c(x,y)$ for each $x \in X_i$ and $y \in X_{i+1}$ for all i

Output: largest set of tuples $(x_1 \dots x_d) \in X_1 \times \dots \times X_d$

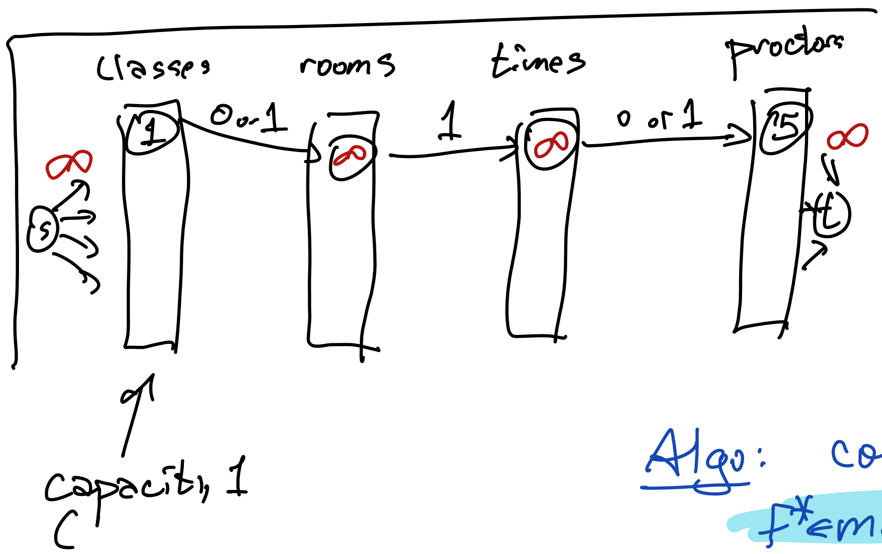
s.t. each $x \in X_i$ appears $\le c(x)$ times
$x \in X_i, y \in X_{i+1}$ $\le c(x,y)$



All constraints must be between adjacent pairs $X_i\ X_{i+1}$

largest valid set of tuples ⇄ largest valid set of paths from s to t ⇄ maximum integer feasible flow from s to t

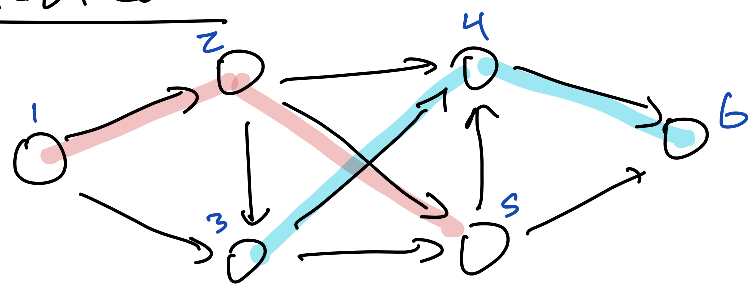
---



Algo: construct G
$f^* \leftarrow \text{maxFlow}(G)$
if $|f^*| < n$
return FALSE
decompose f^* into paths
write each path as tuple

$O(N^3)$ time
N = total input size

Disjoint-Path Cover



Find <u>smallest</u> set of vertex-disjoint paths that cover every vertex of G

Given n envelopes height h_i width w_i

put i inside j $\iff$ $h_i < h_j$ and $w_i < w_j$

Nest envelopes into as few nested sets as possible.

MATCH
Select
Assign

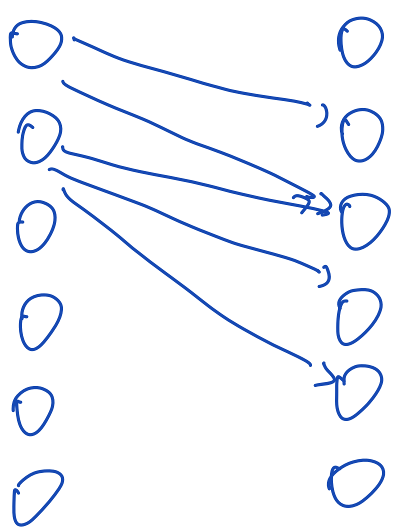
nodes in G to their successors

Every path has a last vertex w/ no succ

min #paths = max #successors

Build bipartite graph $G' = (V', E')$

$V' = L \cup R$ $\quad$ L = copy of V
R = copy of V



"$E' = E$"

$E' = \{(u_L, v_R) \mid u \to v \in E\}$

Find max matching in G'

↓

successors in G

↓

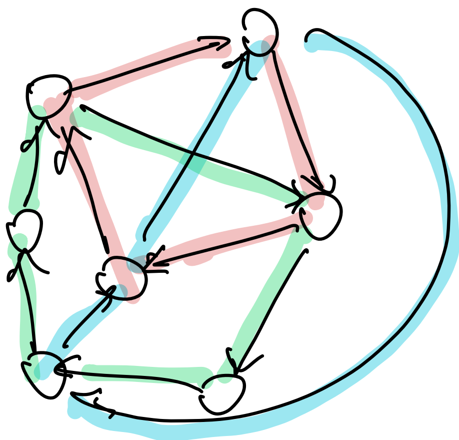
paths in G

#paths in G = #V − #M

matching time:

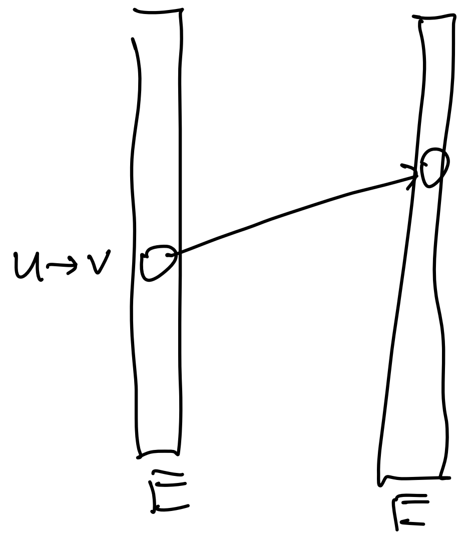
$O(V'E') = O(\underset{n}{V}\underset{n^2}{E})$ $= O(n^3)$ for nesting envelopes

Given a directed graph G <u>not</u> a dag



~~Cycle cover~~
Cover the edges of G with edge-disjoint cycles

For each edge, <u>select</u> its successor on the same cycle



$V' = E \cup E$

$E' = \{(u\to v)\to(v\to w) \mid u\to v \text{ and } v\to w \in E\}$

Perfect ~~Max~~ matching

$O(V'E') = \underline{O(E^2V)}$

$E' \le E^2$ ✓

$E' \le E \cdot V$