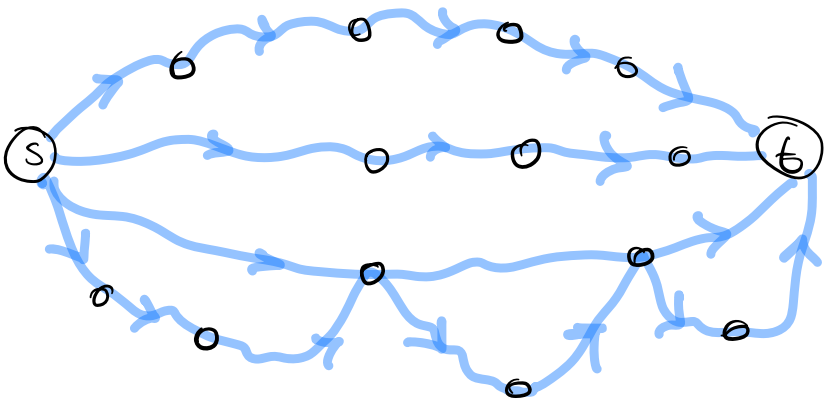
Network Flow Applications cont'd 11.06

See Jeff Erickson's text "Algorithms", Chap 11.

11.1 Edge Disjoint Paths



What is the max number of edge disjoint-paths from s to t in directed graph G.

How to compute it using Network Flow algs:

- give every edge capacity = 1.
- Run Network Flow alg
- the value of the flow is exactly the number of edge-disjoint s-t paths.

Using Orlin's alg, this can be done in $O(nm)$

But Ford & Fulkerson (generally $O(nm^2)$) also runs

(nm iterations, $O(n)$/iteration)

in $O(nm)$ in this case since

- value of flow increases by ≥ 1 each iteration.
- total value is $\leq n-1$

Reduce max # edge-disjoint paths in undirected G

to max # edge-disjoint path in directed G:

Vertices can have capacities.

In addition to the other constraints, each vertex v has a capacity $c(v)$ for flow that can go through v on the way from s to t.

$$\sum_{u \to v} f(u \to v) \leq c(v)$$

Reduce Network-Flow with vertex-capacities to Network Flow (Classic):

The reduction takes $\Theta(m)$ time (in a connected graph, $m \geq n-1$)

$\therefore$ Running time for NF_w vertex-capac is $\Theta(nm)$ (using orlin's alg)

11.4 Tuple Selection.

- Bipartite max matching is a special case of this
- Lets start with an example:

Exam Scheduling

VIU has classes that need exams to be held in rooms and that will be overseen by proctors.

Constraints and Inputs

$E[1 \ldots n]$ where $E[i]$ = # of students in class i

$S[1..r]$ where $S[j]$ = # of seats in room j

A class i can have its exam in room j only if $E[i] \leq S[j]$

$A[1..t, 1..p]$ where $A[k, \ell] =$ TRUE iff k^{th} proctor is available during k^{th} time slot

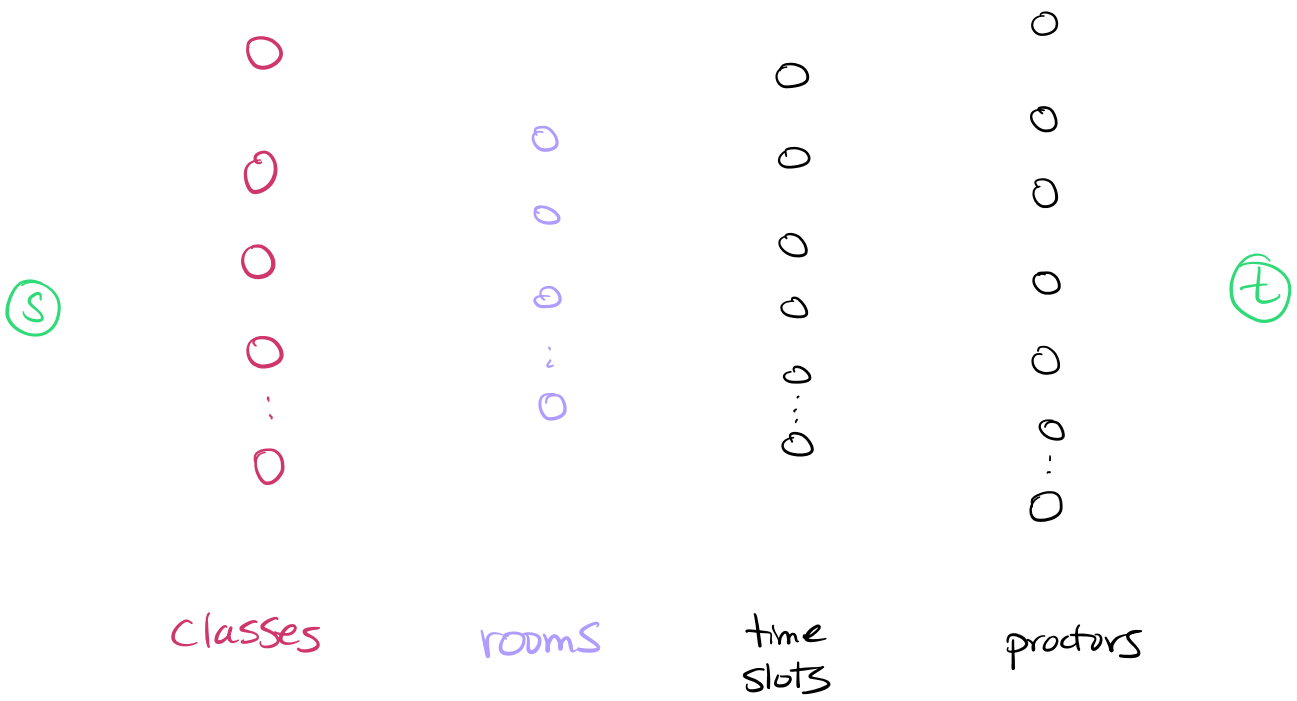
Furthermore, no proctor supervises > 5 exams.

Total size of input is $N = n + r + tp$

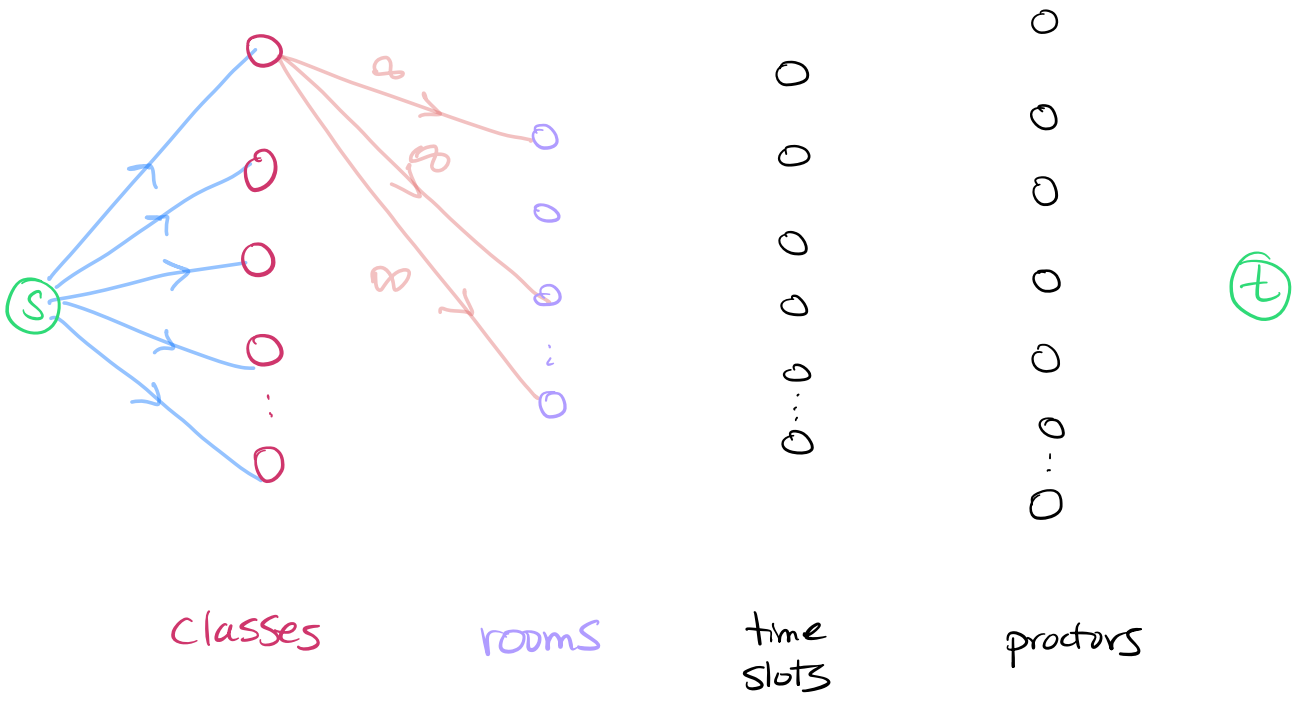
Goal: Schedule classes, rooms, time slots, proctors so as to meet the constraints.
(or say no such schedule is possible)

To solve :

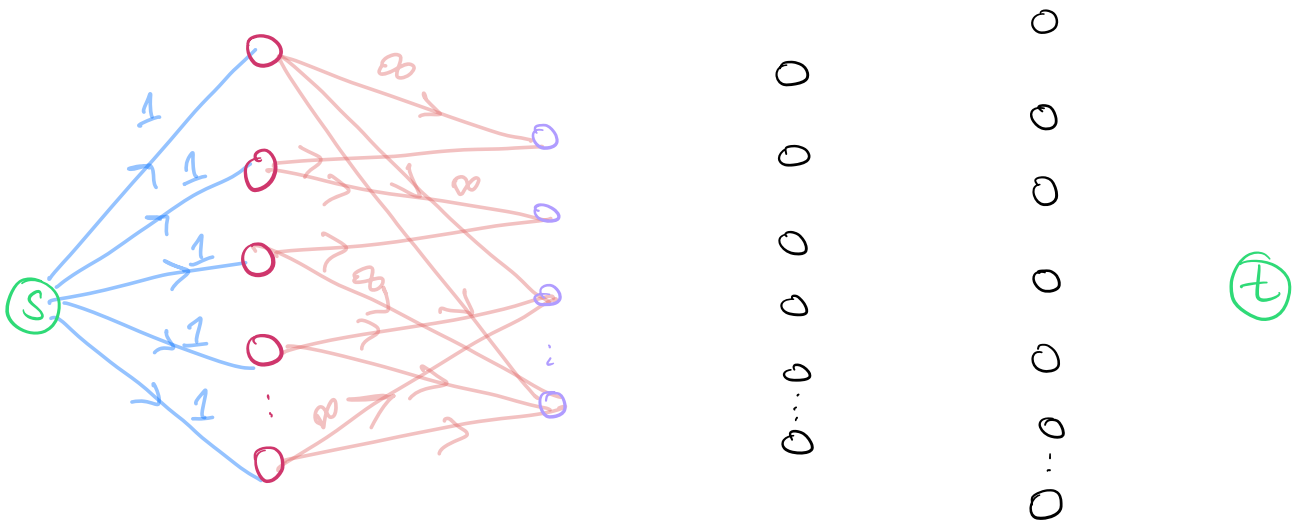
To solve:



To solve:

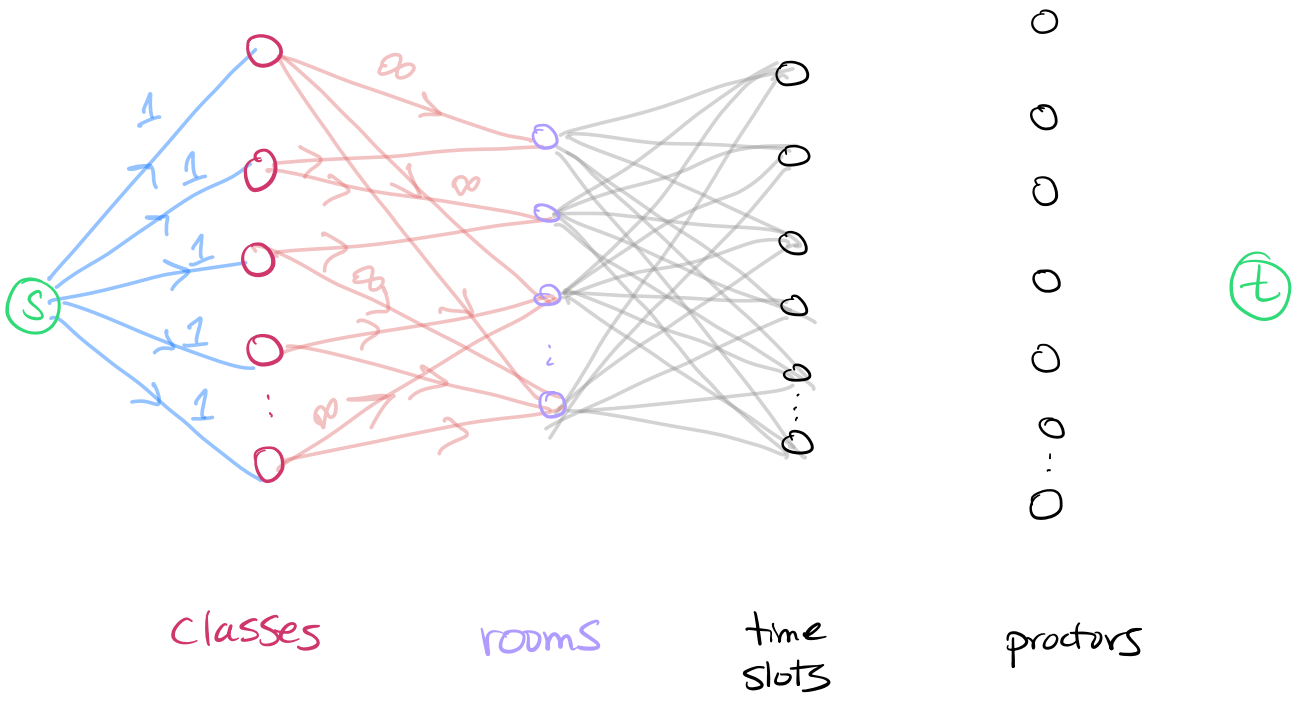


To solve:

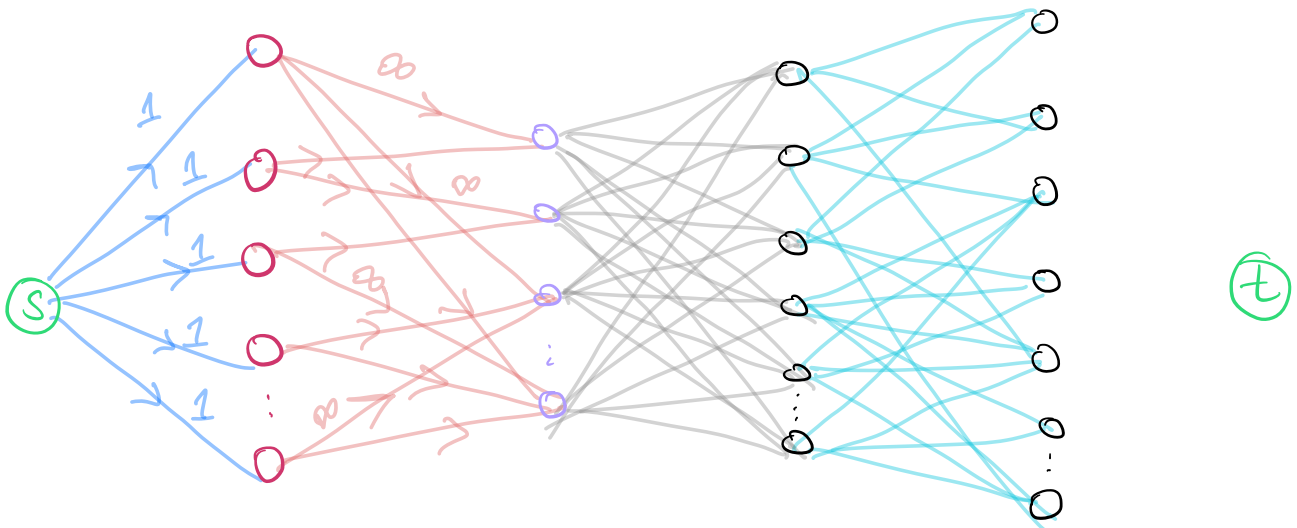


classes rooms time slots proctors

To solve :

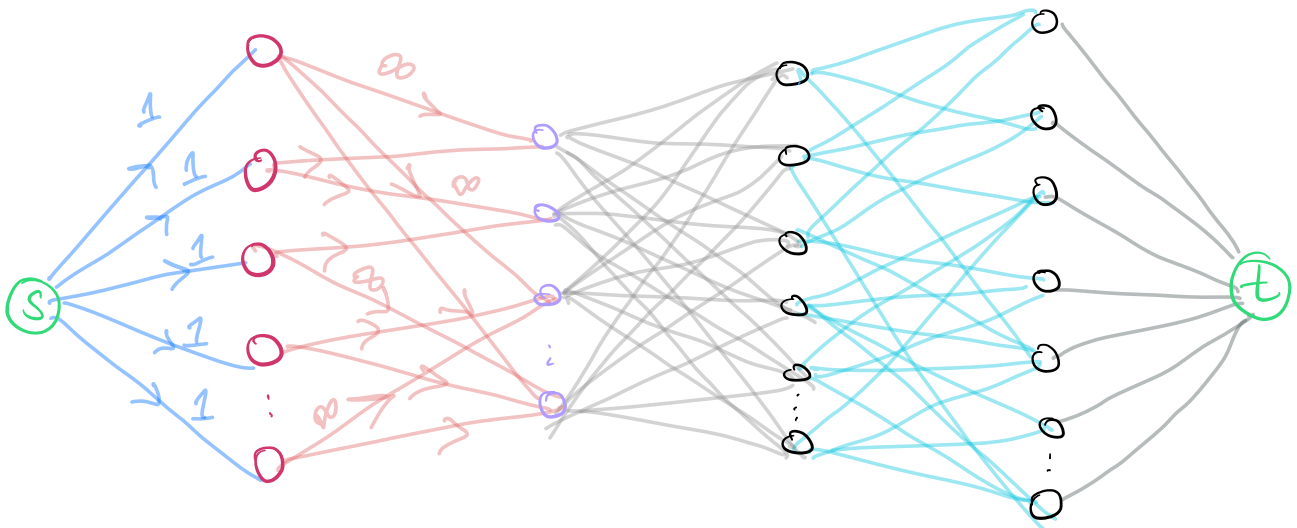


To solve :



classes rooms time slots proctors

To solve:



classes rooms time slots proctors

NF ⇒ Schedule

Suppose ∃ a NF in the network of value _____

1. It assigns each proctor to ≤ 5 time slots, ≤ 1 class per time slot
2. It obeys proctor ↔ time slot constraints.
3. It identifies, for each class, exactly 1 (room, time slot, proctor) triple
4. No room has ≥ 2 classes in same time slot

∴ Follow the flow of 1 unit from a class — it will identify the room + time slot and proctor.

Schedule ⇒ NF

Suppose ∃ a schedule that meets all constraints.
For each (class, room, time slot, proctor) tuple,
push 1 unit of flow through that path in the NF
diagram. It will meet all the flow-capacity constraints
(because we built the NF instance to reflect those
constraints!)

- through each class will be a flow of 1
- ∴ total flow will be n.