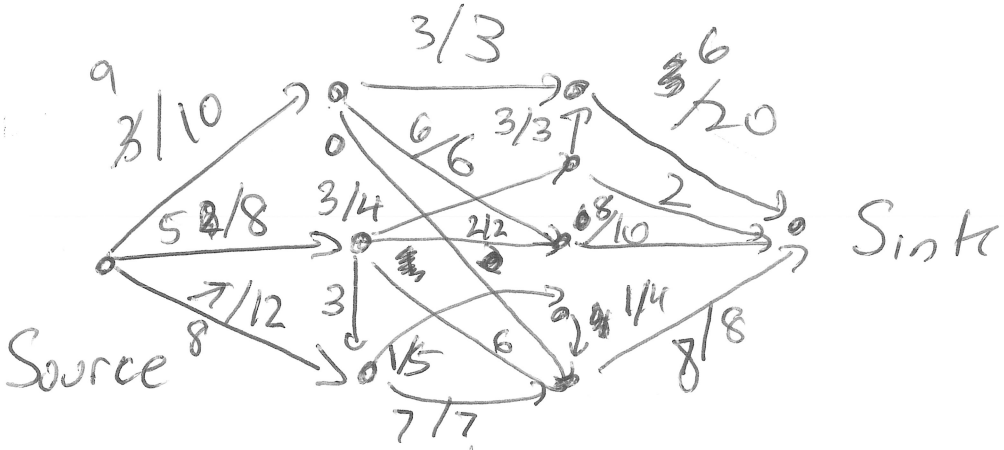
Network Flow



Directed graph

Each has a capacity (max amt of stuff that can flow through per unit of time)

2 special vertices

Source = only outgoing edges

Sink = only incoming edges

$\sum$ flow in = $\sum$ flow out for all but Source, Sink

flow = $\sum$ flow out source = $\sum$ flow in sink

GOAL: What is the max amt of flow we can send from source to sink each unit of time?

HOW TO SOLVE

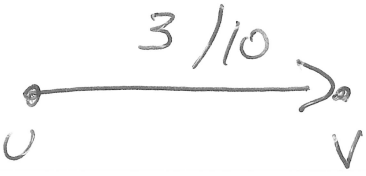
While (there is an augmenting path)
    add max flow you can to that path

NEXT Q

How do I find an augmenting path?

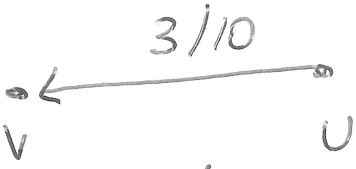
What exactly is an augmenting path?

It is a path from source to sink with linked edge where every edge has open capacity in the appropriate direction.

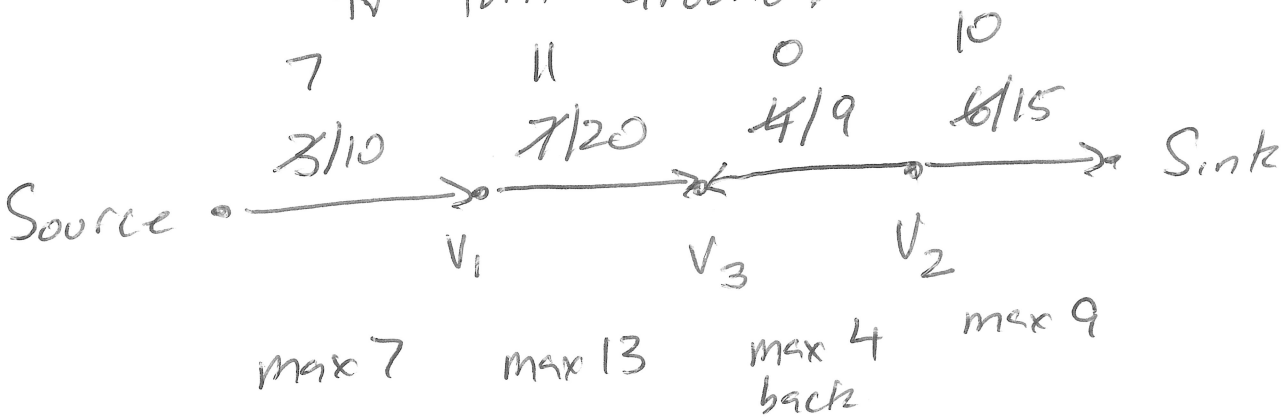


Pic A

I can send $10-3=7$ more units from $U \to V$.



BACK EDGE: I can ALSO send 3 units from V to U (Telling those 3 people to turn around.



Send 4 through

How to find augmenting path?

X (1) DFS (Ford-Fulkerson Alg)

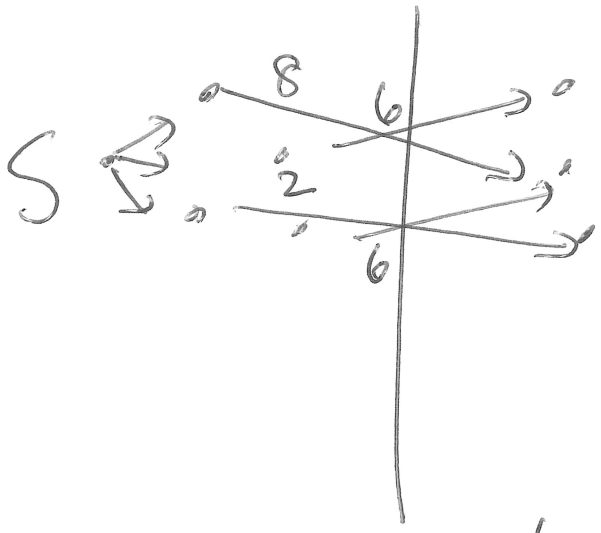
(2) BFS (Edmunds-Karp)

X (3) BFS/DFS (Dinitz's Alg - best in practice)
More Complicated to Code

Code on Site - use as black box

Proof of Correctness

Define a cut as a partition of vertices of the graph into 2 sets where the source (S) and the sink (T) are in different sets.



define value of the cut to be sum of the capacities of the "cross edges" these

2 Teams / 2 Regions

examine all edges from a vertex in S to a vertex T

Max Flow $\leq$ ANY CUT

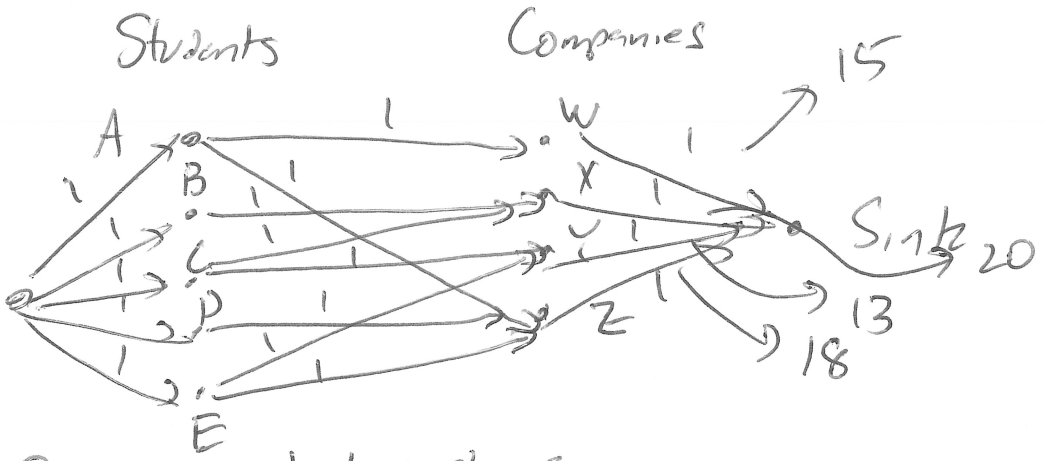
$\Rightarrow$ Max Flow $\leq$ Min Cut

If ~~no~~ there is no augmenting path, we've saturated every possible cut + have "filled" the min cut.

$\Rightarrow$ MAX FLOW = MIN CUT

Applications

Bipartite Matching



Students + Internships

Draw edges w/ capacity 1 for each offer

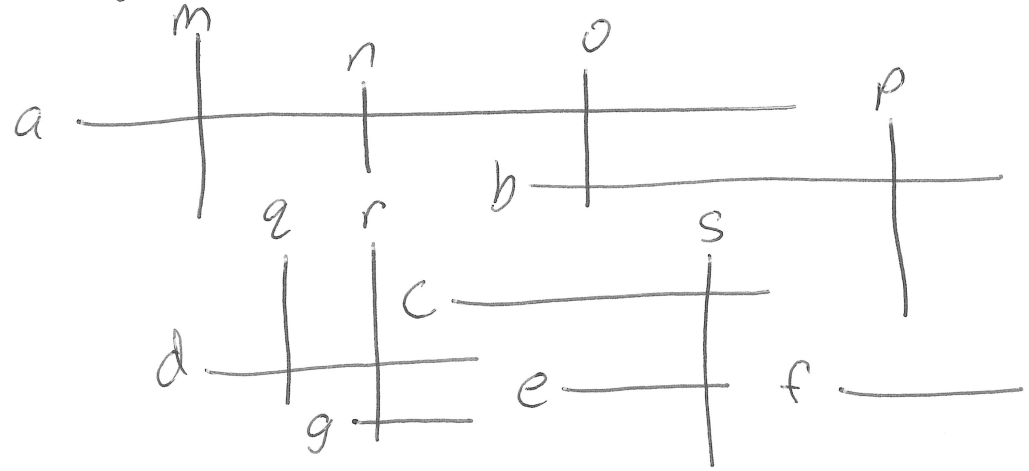
Add edges source each student cap 1.

" " comp each to sink cap 1.

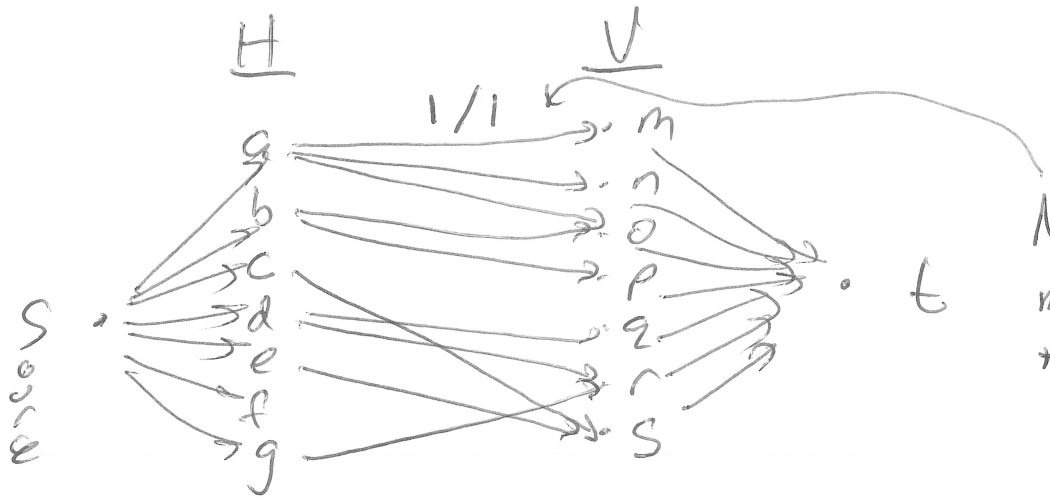
* Note: If companies can take > 1 student it's no longer bipartite matching, but network flow still ~~stou~~ solves the problem.

Min Cut Problem

Input: Some horizontal, vertical line segments



Goal remove fewest # of segments so there is no intersection.



max flow = min # of removals

Ans = (total # lines) - max flow

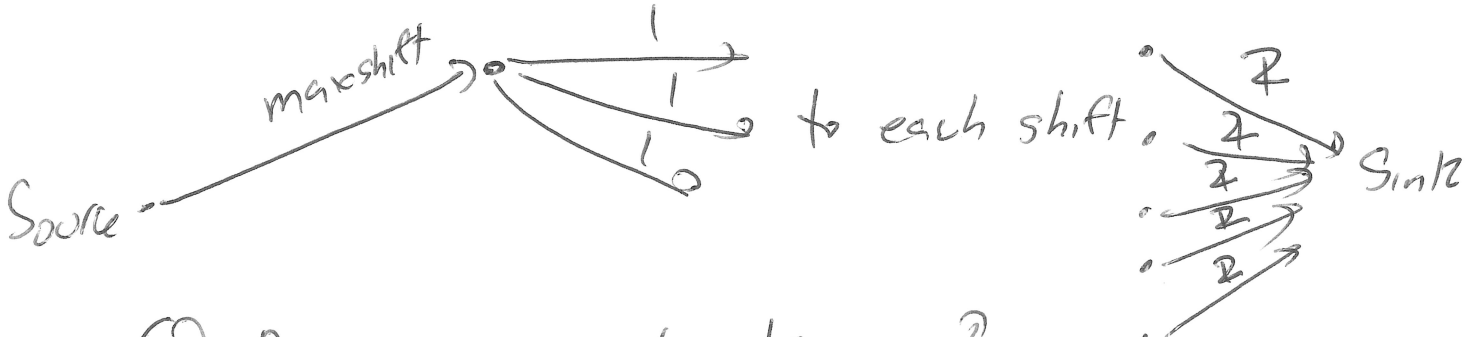
Problem: extra problem Cow Steeplechase

Museum Guards

48 shifts to cover (30 min shift one full day)

Guards to schedule
↳ has max # hours
has some blocked time

Problem what is max number k, such that I can schedule all 48 shifts to have at least k guards each?



(1) Binary search k.

(2) Run flow on this graph to set yes/no
yes = flow == 48*k

TEST

Algorithm Analysis

- summation technique mult sum + sub
- expected value of a discrete random variable (in reference to avg case runtime)
- lower bound comparison sorting

Extra Sorts

- Bucket Sort
- Radix Sort (tracing)

Greedy Algorithms

- Huffman Coding (tracing)

Graphs

- Representation (Adj List Representation)
- DFS
- BFS
- Topological Sort
- Dijkstra's
- Min Span Tree
  - Prim's
  - Kruskal's
- Network Flow
- High Level Setting UP Graph

NO CALC

AIDS: 1 SHEET OF NOTES (Front/Back 8.5" × 11")