Problem 4

Two companies Pineapple and Nanosoft make competing versions of m different software products. Pineapple charges p_i for product i and Nanosoft charges n_i for its version of i. A consumer wants to buy one version of each product. While the customer prefers cheaper versions, she also prefers to buy most software from the same company. In particular, products i and j, if bought from different companies, impose an incompatibility cost of $c(i, j)$ on the customer. The customer's total cost from buying software is the prices paid to the two companies, plus a sum over all incompatible pairs of the respective incompatibility costs. The goal is to determine, for each software product, from which company the customer should buy that product so as to minimize her total cost.

Design and analyze an efficient algorithm for this problem.

m products

↳ p_i

↳ n_i

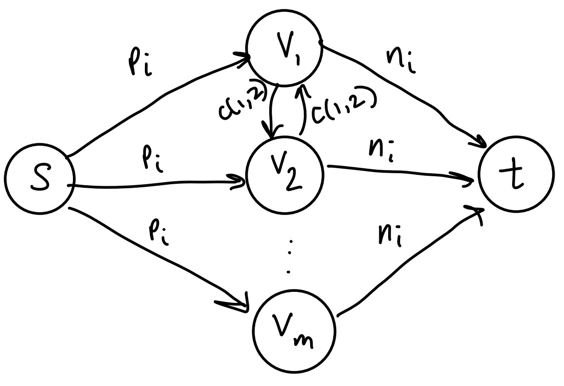
↳ $c(i,j)$

s-t cut

if $v_i \in T$ buy from pinapple

$v_i \in S$ buy from nanosoft

products $v_1 \cdots m$



S T

CS 577 conquer CS department

convince every other class to join their group, but it'll cost. m amount of classes

I_k = cost for the k^{th} class to join our federation

E_k = cost for the k^{th} class if their not in our federation

adjacent classes have issues. So if a class is in federation and the other is not there will be a cost to associate that. If both in fed, they must get along + if both not, it wouldn't be our problem.

$B(i,j)$ = border cost

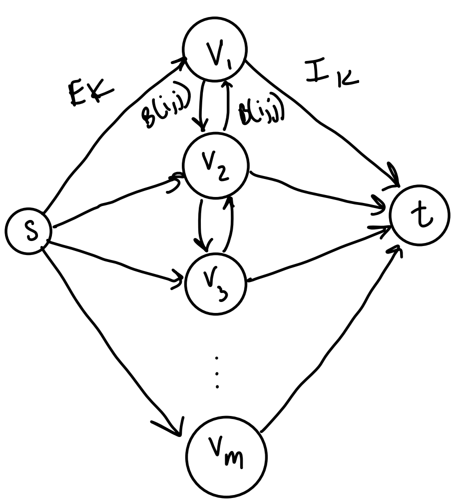
It is also a requirement that if a class is in the fed., all of its prerequisits will be in the fed. as well.

How set up min cut problem?

S-T cut

if $v_i \in S$, then v_i (class) is in federation

if $v_i \in T$, then v_i is not in federation



all classes will have an arrow coming out of it that points to their prerequisite. this arrow will have weight ∞ to represent the requirement

(S) ← CS200 ← CS300 ← CS400

B(i,j) will only be connected for rival adjacent classes

Bipartite vs mincut

CS 577 — Study Group – Network Flow — Fall 2023

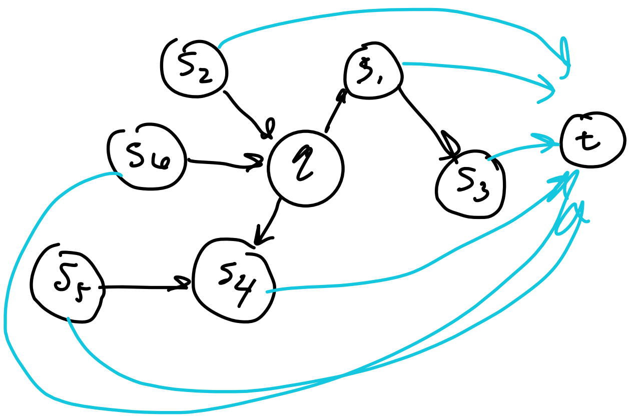
Name: ______________________ Wisc id: ______________________

Basics

1. A spammer is located at one node q in an undirected communication network G and peaceful email users are located at nodes denoted by the set S. Let $c(u, v)$ denote the effort required to install a spam filter for the network edge (u, v). The problem is to determine the minimal effort required to isolate the spammer from the peaceful email users using the spam filters.

Solution:

$s \in G$ $G/S = q$ S/G : all except q



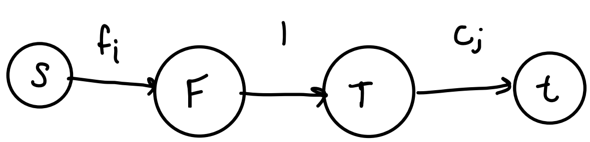
2. Several families go out to dinner together. To increase their social interaction, they would like to sit at tables so that no two members of the same family are at the same table. Assume that there are m families and that the i-th family has f_i members. Assume also that there are n tables, and that the j-th table has a seating capacity of c_j. Show how to find a satisfying assignment of people to tables in polynomial time.

Solution:

find max social interaction

families tables

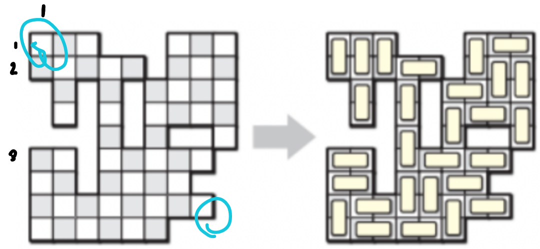
max flow



3. Suppose you are given an $n \times n$ checkerboard with some of the squares deleted. You have a lot of dominos, each of which just the right size to cover two squares of the checkerboard.Describe and analyze an algorithm to determine whether you can cover the board with dominos – each domino must cover exactly two squares, and each square must be covered by exactly one domino.

c_1, r_1
c_2, r_1

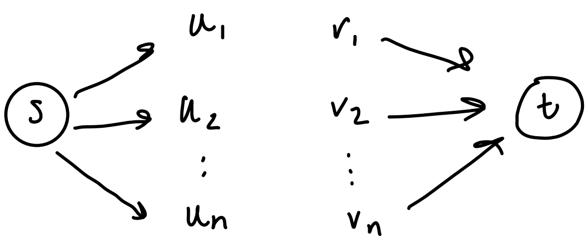
$c_1\ r_1$
$c_1\ r_2$



Solution:

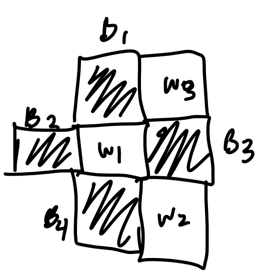
min cut == # of squares on board ∴ yes

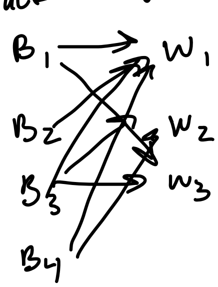
out of all the ways to position dominos



|bip matching| = n

1 or 0





4. Suppose we are given an array $A[1 \cdots m][1 \cdots n]$ of non-negative real numbers. We want to round A to an integer matrix, by replacing each entry x in A with either $\lfloor x \rfloor$ or $\lceil x \rceil$, without changing the sum of entries in any row or column of A. For example:

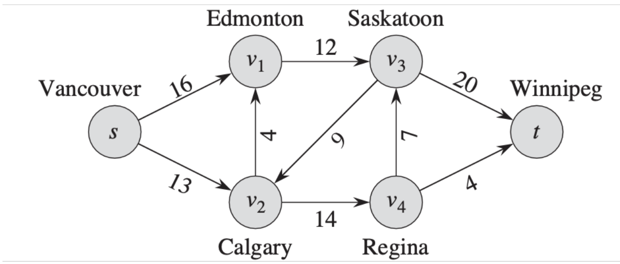
$$\begin{bmatrix} 1.2 & 3.4 & 2.4 \\ 3.9 & 4.0 & 2.1 \\ 7.9 & 1.6 & 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 4 & 2 \\ 4 & 4 & 2 \\ 8 & 1 & 1 \end{bmatrix}$$

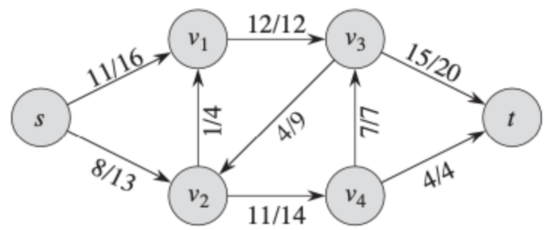
Describe and analyze an algorithm that either rounds A in this fashion, or reports correctly that no such rounding is possible.

Solution:

Name: ____________________ Wisc id: ____________________

Basics

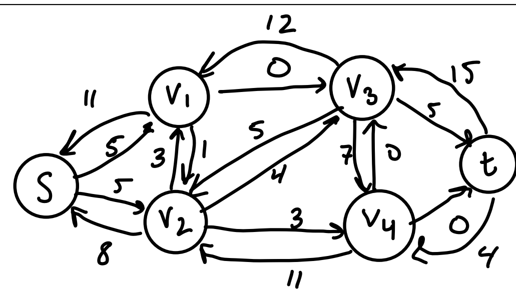




1. *CLRS 3rd edition (p. 717).* The figures above are a flow network G and a flow f. What is the flow value $v(f)$? What is the residual network G_f?

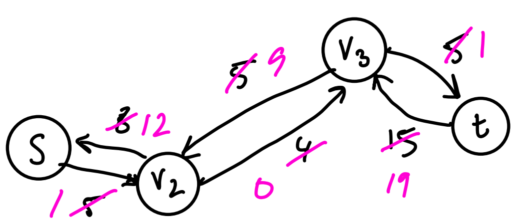
Solution:

15+4=19



2. Consider an augmenting path $p = (s, v_2, v_3, t)$ on G_f. What is `bottleneck(p, f)`? What does the flow network and residual network look like after `augment(f, p)` (increase/augment the flow f along p by `bottleneck(p, f)`)?

Solution:



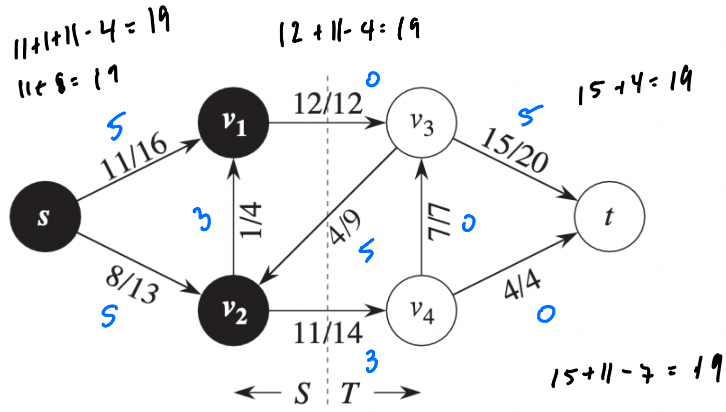
bottleneck

$v_2 \to v_3$

3. Execute basic Ford-Fulkerson algorithm on the graph.

Solution:

$O(\text{maxflow} * E)$



4. Consider a cut (S,T) in G. What is the cut capacity $c(S,T)$? What is the net flow $f(S,T) = \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u)$?

Solution:

net flow = 12 + 11 − 4 = 19

cut capacity = 12+14 − 9 = 17

max flow =

Claim 1. *(CLRS p.721 Lemma 26.4) Let f be any $s-t$ flow and (S,T) be any $s-t$ cut. Then $v(f)$ equals the net flow from S to T. That is, $v(f) = f(S,T)$*

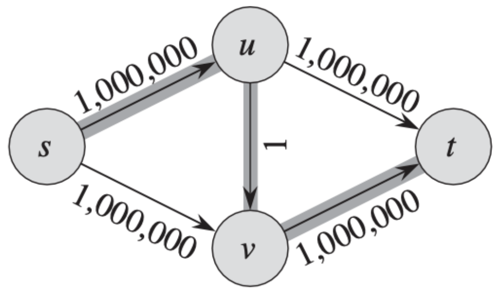
Claim 2. *(CLRS p.723 Lemma 26.5) The value of any flow is bounded above by the capacity of any cut. That is, $v(f) \le c(S,T)$*

Theorem 1. *(CLRS p.723 Lemma 26.6) The following conditions are equivalent*

1. *f is a max flow of G.*
2. *G_f contains no augmenting path.*
3. *$v(f) = c(S,T)$ for some cut (S,T).*

.

Edmonds-Karp



5. What's wrong with running Ford-Fulkerson on the graph above? How does the Edmonds-Karp algorithm improve it?

Solution:

EK prioritizes the shortest path using BFS so it would inevitably

do $s \to v \to t$ instead of $s \to u \to v \to t$ which also

decreases mincut

$O(VE^2)$

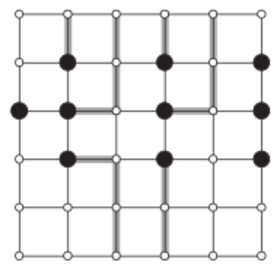
Name: ______________________ Wisc id: ______________________

1. Suppose we have capacities on the vertices as well as the edges. That is, in addition to edge capacity $c(u, v)$, we require that for any vertex v other than source s and sink t, the total flow into v (and therefore the total flow out of v) is at most some non-negative value $c(v)$. How can we compute a maximum flow with these new constraints?

Solution:

More Reductions

2. *CLRS p.760* An $n \times n$ grid is an undirected graph consisting of n rows and n columns of vertices. We denote the vertex in the i-th row and the j-th column by (i, j). All vertices in a grid have exactly four neighbors, except for the boundary vertices,which are the points (i, j) for which $i = 1, i = n, j = 1$ or $j = n$. Given $m \leq n^2$ starting points $(x_1, y_1), ..., (x_m, y_m)$ in the grid, the **escape problem** is to determine whether or not there are m edge-disjoint paths from the starting points to any m different points on the boundary. For example, the grid in the following figure has an escape. Give an algorithm to solve the escape problem. What if we're looking for vertex-disjoint paths?



Solution:

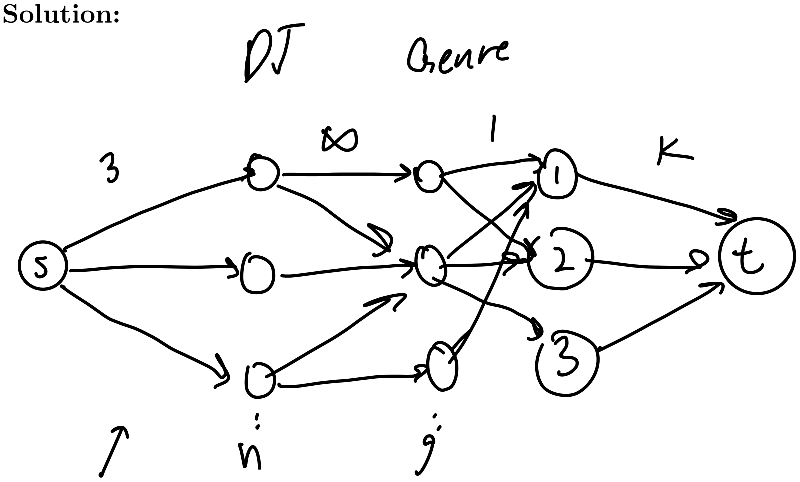
3. You're organizing the First Annual UW-Madison Computer Sciences 72-Hour Dance Exchange, to be held all day Friday, Saturday, and Sunday. Several 30-minute sets of music will be played during the event, and a large number of DJs have applied to perform. You need to hire DJs according to the following constraints.

   - Exactly k sets of music must be played each day, and thus 3k sets altogether.
   - Each set must be played by a single DJ in a consistent music genre (ambient, bubblegum, dubstep, horrorcore, hyphy, trip-hop, Nitzhonot, Kwaito, J-pop, Nashville country, . . .).
   - Each genre must be played at most once per day.
   - Each candidate DJ has given you a list of genres they are willing to play.
   - Each DJ can play at most three sets during the entire event.

   Suppose there are n candidate DJs and g different musical genres available. Describe and analyze an efficient algorithm that either assigns a DJ and a genre to each of the 3k sets, or correctly reports that no such assignment is possible.

Solution:

k each music day



DJ play @ most 3 times

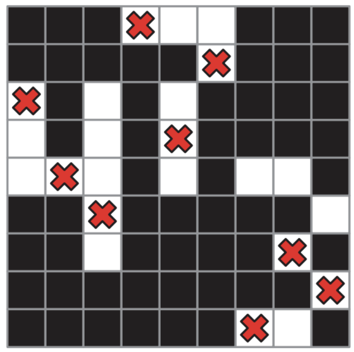
willing to play

flow must be integral

4. Suppose we are given an $n \times n$ square grid, some of whose squares are colored black and the rest white. Describe and analyze an algorithm to determine whether tokens can be placed on the grid so that

   - every token is on a white square
   - every row of the grid contains exactly one token
   - every column of the grid contains exactly one token

   Your input is a two dimensional array $IsWhite[1..n, 1..n]$ of booleans, indicating which squares are white. Your output is a single boolean. For example, given the grid above as input, your algorithm should return True.



Solution:

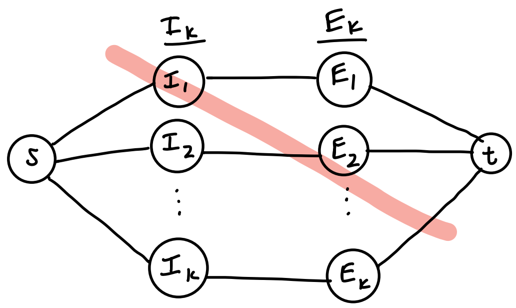
CS 577 conquer CS department

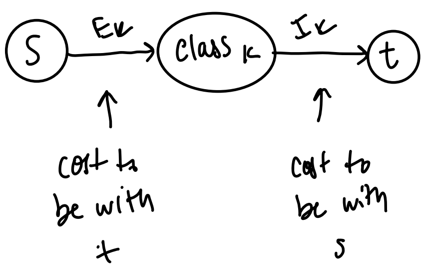
convince every other class to join this group, but it'll cost.

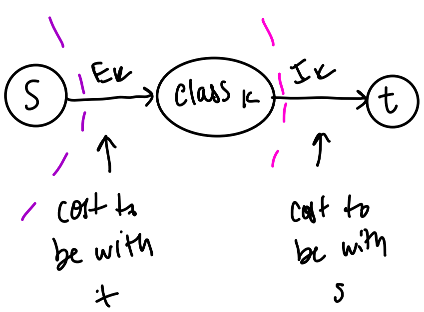
I_k = cost for the k^{th} class to join our federation

E_k = cost for the k^{th} class if their not in our federation

How set up min cut problem?







mincut if $E_k < I_k$

mincut if $I_k < E_k$

X	Y	Z
5	7	7
4	7	7
6	7	7
5	8	7
5	6	7
5	7	8
5	7	6

adjacent have issues

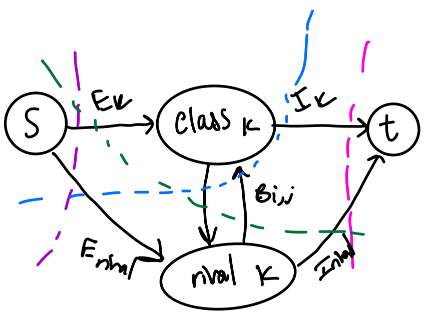
577 → 477 +1

677 → 587 +2

only if 1 is in fed + one not

$B_{i,j}$ cost border

requirements as ∞



both class k + rival not in

both class k + rival in fed

class k in + rival not

rival k in + class not

make sure 577 always in A

∞ cannot be $S \to t$ or min cut $= \infty$

∞ from $s \to x$, x should always be included in S must include prerequisits to include layer classes