Review

A2 $p \vee (\bar{q} \wedge \bar{r})$

$(q \vee r) \rightarrow p$ →

premis

$(q \vee r)$

p

$(q \vee r) \rightarrow p$.

1. $p \vee (\bar{q} \wedge \bar{r})$ premis
2. $q \vee r$ assume
3. p assume — want to get to p
4. $p \rightarrow p$ 3-3, →i
5. $\bar{q} \wedge \bar{r}$ assume — want to get to p
6. q assume — want to get F
7. $\bar{q}$ 5, ∧e
8. F 6,7,
9. $q \rightarrow F$ 6-8, →i
10. r assume
11. $\bar{r}$ 5, ∧e
12. F 10,11
13. $r \rightarrow F$ 10-12
14. F 2,9,13.
15. p →
16. $(\bar{q} \vee \bar{r}) \rightarrow p$ 5-15.
17. P. 1,4,16
18. $q \vee r \rightarrow p$.

Only way you can use ∨ to prove something

$p \vee q$

$p \rightarrow R$

$q \rightarrow r$

$\overline{}$

R

$p = p$

$Q = \bar{q} \wedge \bar{r}$

Know that

$p \rightarrow p$

and $\bar{q} \wedge \bar{r} \rightarrow p$

} Somewhere in the box

then $\overset{\rightarrow F}{q} \vee \overset{\rightarrow F}{r}$

False does not imply anything from a line false, if you don't start boxing, can write anything

---

prove

$p \vee q, \bar{q} \vdash p$

1. $p \vee q$ premis

$\bar{q}$ prem

p as

$p \rightarrow p$ →i

q as

F ¬e

p Fe

$q \rightarrow p$.

p ∨e

Traveling Salesman problem. (TSP)
~~Input~~: A set V of vertices, wheights $w_{uv} \geq 0$ between every pair of unordered vertices.

Output: Minimum weight (cost) hamiltonian cycle

Applications: Visiting a list of cities and come back home while minimizing travel cost.
Drilling many holes ? minimize traveling time (Not drilling)

* NP-complete

Remark: In these applications, the weights satisfy
$w_{u,v} \leq w_{u,x} + w_{x,v} \quad \forall u,v,x \in V$
Known as the triangle inequality: Always cheaper to go directly from one vertex to the other.

Solutions
- Brute Force: try all $n!$ routes
- Heuristics: Solutions that work, but not proven
- Restrict inputs
- Settle for a sub-optimal solution w/ some guarantee on performance

In our case, we will obtain a solution which costs at most twice the cost of the best solution.

Algorithm

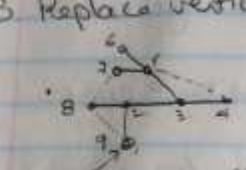
① Find the MST "T" of $\overset{\text{complete graph on n vertices}}{\textcircled{K_n}}$ with input weights w where $n = |V|$.

② Walk "around the tree".

2 (con't)

Each edge used exactly twice

3. Replace vertices visited multiple times with "shortcuts".



Because of triangle inequality, our cost only went down.

Means, cost w/ shortcuts ≤ cost = 2 cost of T

Deleting an edge from a Hamiltonian cycle gives a spanning tree

≤ 2 cost of best Hamiltonian cycle

☒

ex

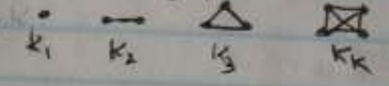
1) Find SP & 2) go around

Defn: A subgraph H on G is a graph with $V(H) \subseteq V(G)$ and the edges of H $E(H) \subseteq E(G)$ with both endpoints in $V(H)$.

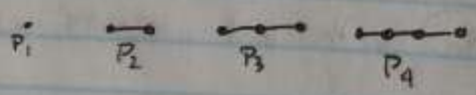
$E(H) \subseteq \{(u,v) \mid (u,v) \in E(G), u \in V(H), v \in V(H)\}$

Special graphs

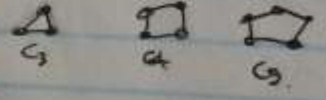
K_n:



P_n:



C_n:



Math 363
pt 18

__Defn__: G contains a path of length k if P_k is a subgraph
" "cycle" " " C_k "

__Defn__:
An __induced__ subgraph H of G is a subgraph with
$E(H) = \{(u,v) \mid (u,v) \in E(G), u \in V(H), v \in V(H)\}$

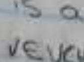
__Defn__: A connected component H of G is a maximal connected subgraph of G.

__e.g__

G= 

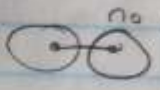
Then  is a subgraph

 is induced

 induced, connected but not maximal

 is a connected component

If $F \cup \{e\}$ has no cycles then



e=(u,v)

Chinese Postman Problem

A. postman starts at a post office and must deliver mail to all houses on all streets and return to the post office. All streets are lined w/ houses ⇒ want to minimize total distance travelled.

__Input__: Connected graph G, weights $w_e \geq 0$ $\forall e \in E$

__Output__: Minimum weight closed walk which contains every edge Or: minimum total weight of edges we need to "double" to make the graph Eulerian

In some cases, we already know the answer when G has an eulerian circuit

Math 363
pz 17.

We need
- Shortest path
- Matching

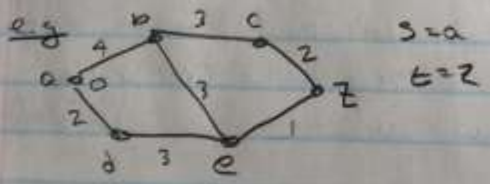
Input: Graph $G=(V,E)$, weights $w_e \geq 0$ $\forall e \in E$ and $s,t \in V$.

Output: Min weight s.t. path.

Alg. Dijkstra's algorithm

```
Initialize an array d indexed by V to $\infty$ (these will be shortest distances from s)
Initialize p[v] = Null
Initialize a set S to $\{s\}$ $d[s] \leftarrow 0$.
While S does not contain t
    Pick $e=(u,v)$ with $u \in S$, $v \in V \setminus S$ which minimizes
        $d[u] + w(u,v)$
        $d[v] = d[u] + w(u,v)$
        $S \leftarrow S \cup \{v\}$
        $p[v] = u$
Return $d[t]$
```

e.g.



$s=a$ let ⊙ = Set S.

$t=z$

Start w/ $S=\{a\}$.

Picks edge not in S w/ min w ⇒ pick (a,d).

$d[d] \leftarrow 2$, $S=\{a,d\}$

Pick b, $d[b]=4$. $S=\{a,b,d\}$.

Pick z, $d[z]=6$. $S=\{a,b,d,z\}$

Math 363
p230

Dijkstra's algorithm

```
Init d
d[s] ← 0
S ← {s}.
Init prev

while $t \neq S$
    Pick $e=(u,v) \in E$ where $u \in S$, $v \in V \setminus S$ minimizing $d[u]+w_{u,v}$
    $d[v] \leftarrow d[u] + w_{u,v}$
    $S \leftarrow S \cup \{v\}$
    prev[v] ← u
return d
```

__Remark__: Can be used to find the lowest weight path to all vertices from s.

Proof of Correctness

__Lemma__: d values are assigned in non-decreasing order

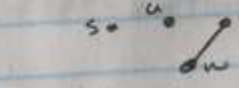
__Pf__: Suppose not. Look at the first time when we assign a d value, say d[v], which is smaller than the one assigned in the previous iteration, say d[u].

Case 1: u=prev[v]

$w_{u,v} > 0$

$d[v] = d[u] + w_{u,v}$

※ $d[v] < d[u]$



Case 2: u≠prev[v]

w∈S and w was in S in the previous iteration

So we could choose the edge w,v in the previous iteration

But we chose an edge w/ endpoint u instead.

\# how we pick the edges.

Math363

P221

Thm: the values d returned by Dijkstra's algorithm are minimum weight distances from s

Pf Suppose it does not

let d^* be the minimum weight distances from s.

Let v be a vertex with lowest d^* value

$d^*[v] < d[v]$.

Let $s, p_2, \ldots, p_{k-1}, v$ be a min weight path from s to v.

$d[p_{k-1}] < d[v]$ since edge weights are >0, $d[p_{k-1}] = d^*[p_{k-1}]$ by our choice of v

$\text{prev}[v] \neq p_{k-1}$

let $u = \text{prev}[v]$

When we set $d[v]$, we could not choose the edge (p_{k-1}, v) as otherwise we would have set $d[v]$ to $d^*[p_{k-1}] + w_{p_{k-1},v} = d^*[v]$

So p_{k-1} was not in S.

This means we set the value of $d[p_{k-1}]$ after we set the value of $d[v]$.

※ to lemma ■

We've actually seen a simpler version of Dijkstra's algorithm. Normally, it uses a priority queue to speed up picking the edge in each iteration

Bipartite matching

Recall: a <u>biparte graph</u> is a graph $G = (V, E)$ such that $V = V_1 \cup V_2$ and there are no edges with both endpoints in V_1 or V_2.

A <u>matching</u> M in G is a subset of the edges such that each vertex is the endpoint of at most one edge in M.

Math 263
P222

M-augmenting path is a path with edges alternating between being in M and not in M and the first and last edge are not in M

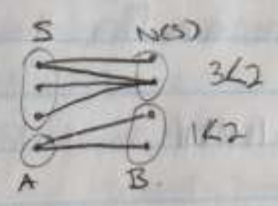
Algorithm: Start with ~~Se~~ $M=\emptyset$. Find an M-augmenting path P. Switch edges of M along this path

Biparaite Matching Algorithm.
- Start w/ $M=\emptyset$
- Find an M-augmenting path P
- Swap on P
- Repeat until no more augmenting paths

Thm: M is a maximum matching $\Leftrightarrow \nexists$ an M-augmenting path.
Defn: Perfect Matching M is a matching where every vertex is the endpoint of some edge in M.

Hall's theorem. Let G be a bipartite graph w/ a partition (A,B). G has a perfect matching $\Leftrightarrow \forall S \subseteq A, |S| \leq |N(S)|$ and $|A|=|B|$
$N(S)=\{u \mid \exists (u,v) \in E(G), v \in S\}$.

e.g



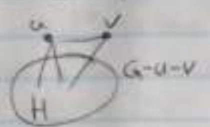
N(S) = Neighborhood of S

Pf: Suppose the theorem is false. Let G be a counter-example (bipartite, all [illegible] but no perfect matching) minimizing the size of either side (|A|)
|A|>0 since the graph w/ no vertex has a perfect matching

Math 263
PZ 23.

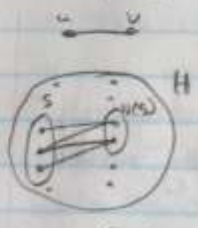
$|A|>0$, $|E(G)|>0$ otherwise $0=|N(\{v\})|<|\{v\}|=1$ for any vertex v

Let $e=(u,v)\in E(G)$ $u\in A, v\in B$.



Look at $G-u-v=H$. H is still bipartite since we removed 2 vertizes.
is $|S|\leq|N(S)|$ $\forall S\subseteq$ first bipartition of H still true?
If yes, by minimality, h would have a perfect matching, say M_0. But then if we add $e=(u,v)$, $M_0\cup\{(u,v)\}$ is a perfect matching in G, #

If no, $\exists S\subseteq$ first bipartition of H s.t $|S|>|N(S)|$ in H.



The only thing G has more than H, only extra G has is u & v, but neither are in the neighborhood of S
In G, $N(S)=\{v\}\cup N_H(S)$ Neighborhood of S in H

Let G_1 be the graph with vertex set $S\cup N_G(S)$ and all edges between those vertices.
Let G_2 be the graph with the remainding vertizes $V(G)\setminus[S\cup N_G(S)]$ and all edges between those vertizes
$u\notin V(G_1)$ and $v\notin V(G_2)$ So both graphs are smaller than G

The conditions $\forall S\subseteq$ first bipartition $|T|\leq|N(T)|$ is satisfied for G_1 since $N_G(S)=\{v\}\cup N_H(S)$
The conditions $\forall T\subseteq$ first bipartition $|T|\leq|N(T)|$ is satisfied for G_2 since $N_{G_2}(B\setminus N_G(S))\subseteq A\setminus S$ (no edges btw non-neighbors of S and S)

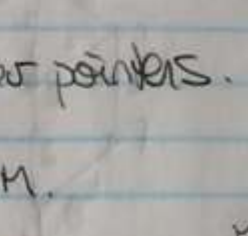
Math 263
p224.

By minimality, G_1 has a perfect matching and G_2 has a perfect matching.
The union of those perfect matching is a perfect matching in G. ■

Hall's Thm: let G be a bipartite graph with parts A and B.
G has a perfect matching $\Leftrightarrow$
$|A|=|B|$ and $\forall S \subseteq A \quad |S| \leq |N(S)|$

Algorithm 1:
Input: $G=(V,E)$ a bipartite graph
M a matching in G
U unmatched vertices in A
W unmatched vertices in B.
Output: Either
① A M-augmenting path
② A set $S \subseteq A$ of vertices with $|S| > |N(S)|$

```
Init prev
  $S \leftarrow U$
  $T \leftarrow \emptyset$
  for $s \in S$ set prev[$s$] $\leftarrow$ Null
  while true
    If $\exists e=(u,v) \in E$ with $u \in S$, $v \notin T$
      prev[$v$] $\leftarrow u$
      If $v \in W$ then
        return path from v folowing prev pointers.
      $T \leftarrow T \cup \{u\}$
      $S \leftarrow S \cup \{$vertex matched to $v$ in $M$.
  Else   prev[$\longrightarrow$] $\leftarrow v$
    return S.
```



Math 263
p225.

Remark: At the beginning of every iteration

- $|T| < |S|$
- $T \subseteq N(S)$
- $T = N(S)$ only if $\nexists e = (u,v) \in E$ with $u \in S$, $v \in T$

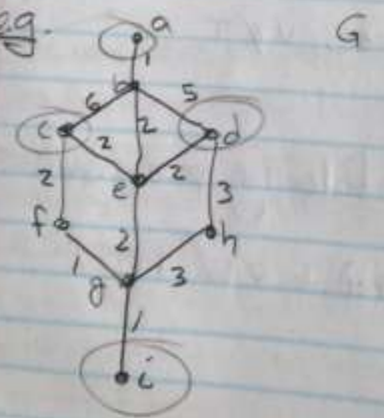
For finding a maximum matching
Alg2. Build a digraph H with vertex set A and directed edges $\{(u,w) \mid \exists v \in B, (u,v) \in M, (v,w) \in M\}$. in H
Then Run a search algorithm (e.g DFS or BFS) starting from U and see if we can reach a vertex in $N(w)$
if we can, translate the directed path we found in H to a M-augmenting path in G.

Chinese Postman Problem

Alg:

1. Compute degrees in G
2. $S \leftarrow \{v \mid v \in V, \deg(v) \text{ is odd}\}$
3. Build H, the weighted complete graph w/ vertex set S and weight $w_{(u,v)}$ = shortest path distance from u to v in G
4. Find a min weight max matching M in H
5. Return all edges of G in paths corresponding to edges of M.

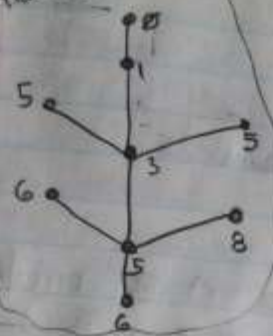
eg.



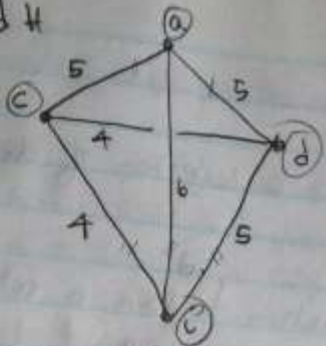
odd vertices:
$S \Leftarrow \{a, c, d, i\}$

Math263
p726

shortest path from a to c,d,i

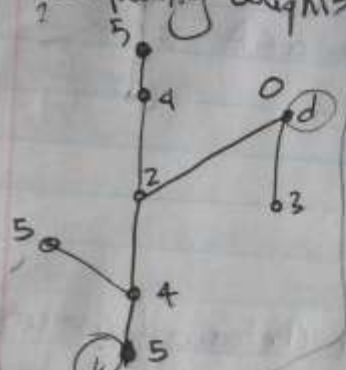


Build H

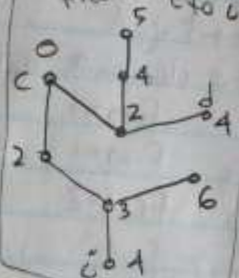


Need shortest path to/from all odd vertices.

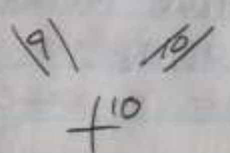
Computing weights from d to i



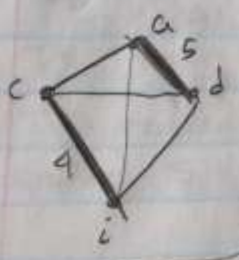
From c to i and d



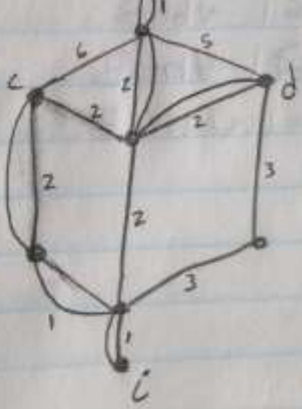
4 Matchings.



Minimum weight perfect matching in H are the edges we need to double



Finally translate to original graph



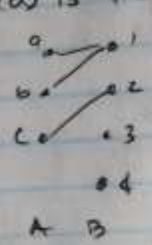
Math 263
PZ27

Counting

Defn. A set is a collection of distinct elements.
e.g $\{1,2,3\}$.

Defn: a function f from a set A to a set B, denoted $f: A \mapsto B$ is an assignment of one element of B to each element of A
$f(a)$ is the element of B assigned to $a \in A$.

eg



$f(a)=1$ $A=\{a,b,c\}$.
$f(b)=1$ $B=\{1,2,3,4\}$.
$f(c)=2$

Remark: We can represent a function (between finite sets) as a bipartite graph where every vertex in A has degree exactly one.

Defn: A function $f: A \to B$ is said to be injective (or one-to-one) if $\forall a_1, a_2 \in A$, $f(a_1)=f(a_2) \Rightarrow a_1, a_2$.
A function $f: A \to B$ is said to be surjective (or onto) if $\forall b \in B \exists a \in A$ s.t. $f(a)=b$
A function f is bijective if f is injective and surjective.
Such an f is a bijection.

Remark: in terms of graphs,
injective $\Leftrightarrow$ $\deg(b) \leq 1$ $\forall b \in B$
surjective $\Leftrightarrow$ $\deg(b) \geq 1$ $\forall b \in B$
bijective $\Leftrightarrow$ $\deg(b) = 1$ $\forall b \in B$.

Math263
p228.

Theorem: If there is a bijection between A and B then $|A|=|B|$

Pf: Follows from Hall's thm from example

e.g. Number of subsets of a set S of size n is 2^n

e.g. $S=\{a,b,c\}$

Subsets $\{\underset{0}{\emptyset}, \underset{1}{\{a\}}, \underset{2}{\{b\}}, \underset{3}{\{c\}}, \underset{4}{\{a,b\}}, \underset{5}{\{a,c\}}, \underset{6}{\{b,c\}}, \underset{7}{\{a,b,c\}}\}$

f: Subsets of $S \to \{0,1,\dots,2^n-1\}$

defined as

$\forall T \subseteq S \quad f(T) = \sum_{s_i \in T} 2^{i-1}$

f is surjective since if $k \in \{0,1,\dots,2^n-1\}$ Then we can write k in binary and $f(T)=k$ if T is $\{s_{i_1}, s_{i_2}, \dots, s_{i_j}\}$ where $i_1, \dots, i_j$ are the bits of k set to 1.

f is injective since $\sum_{s_i \in T_1} 2^{i-1} = f(T_1) = f(T_2) = \sum_{s_i \in T_2} 2^{i-1}$

and these two sums are equal only if every term in the sums is equal

▨

e.g. let S_1 and S_2 be finite sets (Say $S_1=\{0,1,\dots,|S|-1\}$, $S_2=\{0,1,\dots,|S_2|-1\}$)

$T=\{(s_1,s_2) \mid s_1 \in S_1, s_2 \in S_2\}$

T is called the Cartesian product of S_1 and S_2 and is denoted $S_1 \times S_2$

Claim: $|T| = |S_1| \times |S_2|$

Pf $f: T \to \{0, 1, \ldots, |S_1| \times |S_2| - 1\}$ (bijection)

$\forall (s_1, s_2) \in T,\ f((s_1, s_2)) = s_2 + s_1|S_2|$

just left to prove that it is bijective & surjective

example $f(2,1) = 1+2\times2=5.$

f is injective since

$f(s_1, s_2) = f(t_1, t_2)$

$s_2 + s_1|S_2| \quad t_2 + t_1|S_2|$

$$\left\lfloor \frac{k}{|S_2|} \right\rfloor = \left\lfloor \frac{\overset{k}{s_2 + s_1|S_2|}}{|S_2|} \right\rfloor = \left\lfloor \frac{s_2}{|S_2|} + s_1 \right\rfloor = s_1$$

$$\left\lfloor \frac{k}{|S_2|} \right\rfloor = \left\lfloor \frac{t_2 + t_1|S_2|}{|S_2|} \right\rfloor = t_1$$

Shows that $s_1 = t_1$

Since $s_2 + s_1|S_2| = t_2 + t_1|S_2|$

$s_2 = t_2$

Therefore, $(s_1, s_2) = (t_1, t_2)$

f is surjective since

514
287-1881

Let $k \in \{0, 1, \ldots, |S_1||S_2| - 1\}$.

Then $\lfloor \frac{k}{|S_2|} \rfloor \in S_1$ and $k \bmod |S_2| \in S_2$

So $f(\lfloor \frac{k}{|S_2|} \rfloor, k \bmod |S_2|) = k \bmod |S_2| + \lfloor \frac{k}{|S_2|} \rfloor |S_2| = k$

Therefore f is a bijection and the size of T is equal to the size of $\{0, 1, \ldots, |S_1||S_2| - 1\}$.

▨

ex $S_1 = \{0, 1, 2\}$ $\quad S_1 \times S_2 = \{00, 01, 10, 11, 20, 21\}$

$S_2 = \{0, 1\}$

$f(2,1) = 1 + 2\times2 = 5.$

$|S_2| = 2 \quad k = 3 \quad \lfloor \frac{3}{2} \rfloor = 1$

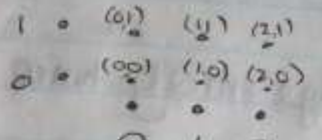
$3 \bmod 2 = 1$

$f(1,1) = 3.$

Math 263
P730.

This is a very formal proof of the intuitive idea that
- we have $|S_1|$ choices for the first coordinate
- regardless of our first choice, we have $|S_2|$ choices for the second coordinate (and diff choices give diff. elements of $S_1 \times S_2$ and there are choices we can make to get every element of $S_1 \times S_2$)

Last time:



$|S_1 \times S_2| = |S_1| \times |S_2|$

<u>Thm</u>: the complete graph on n vertices has $\frac{n(n-1)}{2}$ edges.

<u>Pf</u>: Label the vertices of $K_n = (V,E)$ by $\{0,1,\dots,n-1\}$.

Define $f: E \times \{0,1\} \to V \times V$

$f((u,v),0) = (u,v)$

$f((u,v),1) = (v,u)$

f is injective since if the second coordinate differ, f maps those elements to different elements of $V \times V$

$$\underset{(u_1,v_1)}{\underset{\shortparallel}{f((u_1,v_1),0)}} = \underset{(u_2,v_2)}{\underset{\shortparallel}{f((u_2,v_2)0)}}$$

$\Rightarrow u_1 = u_2$ and $V_1 = V_2$

So $(u_1,v_1) = (u_2,v_2)$ as edges f is surjective

Suppose $(u,v) \in V \times V$

if $u < v$ then $f[(u,v),0] = (u,v)$ and $(u,v) \in E$ since K_n contains all edges

if $u > v$, then $f((u,v),1) = (u,v)$

if $u = v \Rightarrow$ loop $\otimes$ f; nothing maps to identical elements.

Math 263
PZ 31

But if we consider f as a function from $E \times \{0,1\}$ to $V \times V - \{(v,v) \mid v \in V\}$ than f is surjective

So (new) f is bijective

$|V \times V| = |V|^2 = n^2$ and $|\{v,v\} \mid v \in V\}| = n$.

So $2|E| = n^2 - n$ and $|E| = \frac{n^2-n}{2}$

Alternative pf:

pick one vertex (n choices)

pick a different vertex (n-1 choices)

If we do this, we've picked every edge exactly twice.

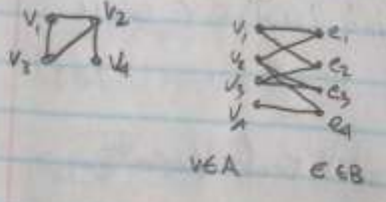
Thm: Let $G = (V,E)$ be a bipartite graph with parts A and B then

$$\sum_{v \in A} \deg(v) = |E| = \sum_{v \in B} \deg(v)$$

Pf: Since every edge has one endpoint in A and one endpoint in B, then $\sum_{v \in A} \deg(v)$ counts every edge exactly once. (via their endpoints in A) And symetrically for B. ▨

Corolary: Let $f: A \to B$ be a function such that every element of B is assigned exactly k elements of A Then $k|B| = |A|$

Pf of handshaking lemma from this.



Math 263
PZ32

Hat's Problem

8 ppl enter a restaurant & leave their hat at the front, then take a random hat when they leave.

1) How many pairings of people w/ hats can you get?

2) How many ways can they leave such that no one has their own hat?

P is set of 8 people.
$f: P \to P$
$f(p) = q$ if person p took q's hat.

counting all bijections.

1) Here, we are counting bijections. These bijection (from a set to itself) are called permutations.

<u>Thm</u>: The number of permutations $f: S \mapsto S$ where $|S| = n$ is $n!$

pf: Let $S = \{s_1, \ldots, s_n\}$. Then we have n choises for $f(s_1)$.
Given the above, we always have n-1 choices for $f(s_2)$
[...]
we have 1 choise left for $f(s_n)$

▨

2) We are actually going to try to count the number of ways in which at least one person gets their own hat, (then take 8! - this #)

Let S_i = {ways where the i^{th} person gets their own hat}

$\sum_{i=1}^{8} |S_i| = 8 \cdot 7!$ * this is overcounting some of the "ways".

For example, the case everyone gets their own hat is counted 8 times.

Math 263
pg 33.

All overcounted ways have at least 2 people got their own hat

$\sum_{i=1}^{8}|S_i| - \sum_{i=1}^{8}\sum_{j=1}^{i-1} |S_i \cap S_j|$ But this still incorrectly counts the cases where at least 3 people have their own hat.

$$\sum_{i=1}^{8}|S_i| - \sum_{1\le i\le j\le 8} |S_i \cap S_j| + \sum_{1\le i\le j\le k\le 8} |S_i \cap S_j \cap S_k| - \dots - |S_1 \cap S_2 \cap \dots \cap S_8|$$

Here, what we used is.

<u>Thm</u>. Inclusion-Exclusion principle for $\forall n$ sets $S_1, \dots, S_n$

$$|S_1 \cup \dots \cup S_n| = \sum_{i=1}^{n} |S_i| - \sum_{1\le i<j\le n} |S_i \cap S_j| + \sum_{1\le i\le j\le k\le n} |S_i \cap S_j \cap S_k| \dots$$

$$+ (-1)^{n+1} |S_1 \cap S_2 \cap \dots \cap S_n|$$

$$\left|\bigcup_{i=1}^{n} S_i\right| = \sum_{i=1}^{n} (-1)^{i+1} \sum_{\{j_1,\dots,j_i\}\subseteq\{1,\dots,n\}} \left|\bigcap_{k=1}^{i} S_{j_k}\right|$$

ex for 2 sets.

	$\lvert S_1\cup S_2\rvert =$	$\lvert S_1\rvert +$	$\lvert S_2\rvert -$	$\lvert S_1\cap S_2\rvert$
$x\in A\cap B$	1	1	1	1
$x\in A\neg B$	1	1	0	0
$x\in B\neg A$	1	0	1	0
other	0	0	0	0

For 3 sets.

$$\left|\bigcup_{i=1}^{3} S_i\right| = |S_1 \cup S_2 \cup S_3| = |S_1| + |S_2| + |S_3| - |S_1\cap S_2| - |S_2 \cap S_3| - |S_1 \cap S_3| + |S_1 \cap S_2 \cap S_3|$$

Math263
p231

Combinations w/ repetitions.

__Q1__: How many ways are there of selecting 7 fruits from a bowl of apples, oranges and pears. (Bowl is ∞ & order does not matter)

__Q2__: How many solutions are there to $x_1 + x_2 + x_3 = 7$ where all variables take non negative integer values.

q1 ⇒ Given a combination, always write the apples first, then the oranges then the pairs.
So for each choice of fruit, there is exactly one way of writing it.

Split the fruits into boxes of same type.
aopaopa ⇒ aaa|oo|pp. (keep separator even if empty category
Because of this, we can identify fruit based on its location relative to splitters.
__ex__ |x|xxxxxx ⇒ ∅a, 1o, 6p.
⇒ Always have 7 fruits w/ 2 splitters, length = 9.

$$\text{So } \frac{\overset{\text{possibilities}}{\boxed{9!}}}{\boxed{2}\boxed{7!}} = \binom{9}{2}$$

=|| 7x

More generally
__Thm__: The number of ways of choosing k elements out of n elements (types) when repetition is allowed is.

$$\binom{n+k-1}{k-1}$$

what we have used implicitly is.
__Thm__: If $f: A \to B$ and $g: B \to A$ are functions and $\forall a \in A$, $g(f(a)) = a$ and $\forall b \in B$, $f(g(b))$, then f and g are bijections.
⇒ Given a function, if you can find an inverse, you have a guarantee that both sets have the same size.

Math263
P237

"Alternative" pf of Q2:
Let $y_j = \sum_{i=1}^{j}(x_i+1) = j + \sum_{i=1}^{j} x_i$

Then $\{y_1, \dots, y_{k-1}\}$ is a subset of $\{1, \dots, n+k-1\}$ and we get every subset exactly once since $y_1 < y_2 < \dots < y_{k-1}$
$y_k = n+k$ (always)

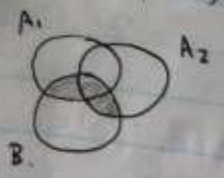
Thm: Every graph of 2+ vertices has at least 2 vertices of the same degree.
Pf: In a graph on n vertices, the vertices can have degrees in $\{0,1,2,\dots,n-1\}$ This set has size n. But a graph can not have both an edge of degree 0 and another one of degree n-1 at the same time (Otherwise, vertex w/deg n-1 would need to be connected to the one of degree 0) So in # graph, vertices have at most n-1 distinct degrees. But there are n vertices, so 2 of them need to share the same degree, by the pigeonhole principle. ▨

Thm (pigeonhole principle) If we put more than n objects into n boxes, then there is a box containing at least 2 objects

Math263
p738

lemma: $(A_1 \cap B) \cap (A_2 \cap B) = A_1 \cap A_2 \cap B$.

Pf



lemma: $(A_1 \cap B) \cup (A_2 \cap B) \cup \ldots \cup (A_k \cap B) = (A_1 \cup A_2 \cup \ldots \cup A_k) \cap B$.

Pf: Let S be the set on the LHS, and T be the set on the RHS.
If $x \in S$, Then $\exists i$ s.t. $x \in A_i \cap B \Rightarrow x \in A_i$ and $x \in B$.
Since $A_i \subseteq A_1 \cup A_2 \cup \ldots \cup A_k \Leftrightarrow x \in A_1 \cup A_2 \cup \ldots \cup A_k$ and $x \in B \Leftrightarrow x \in T$.

Inclusion Exclusion principle Theorem Pf.
Already proved Base case $n=2$.

Suppose for any $n-1$ sets the thm is true
Then for any n sets $S_1, S_2, \ldots, S_n$, $|S_1 \cup S_2 \cup \ldots \cup S_n| = |\overbrace{(S_1 \cup \ldots \cup S_{n-1})}^{A} \cup \overbrace{S_n}^{B}| =$

$$\underbrace{|S_1 \cup \ldots \cup S_{n-1}|} + |S_n| - |(S_1 \cup \ldots \cup S_{n-1}) \cap S_n|$$

$$= \sum_{i=1}^{n-1} |S_i| - \sum_{1 \le i \le j \le n-1} |S_i \cap S_j| + \sum_{1 \le i \le j \le k \le n-1} |S_i \cap S_j \cap S_k| - \ldots + (-1)^{n-1} |S_1 \cap S_2 \cap \ldots S_{n-1}| + |S_n|$$

$$- |(S_1 \cap S_n) \cup (S_2 \cap S_n) \cup \ldots \cup (S_{n-1} \cap S_n)|$$

Apply our inductive hypothesis to the last term of this sum (it is the size of the union of n-1 sets)

$$= \sum_{i=1}^{n-1} |S_i| - \sum_{1 \le i \le j \le n-1} |S_i \cap S_j| + \sum_{1 \le i \le j \le k \le n-1} |S_i \cap S_j \cap S_k| - \ldots + (-1)^{n-1+1} |S_1 \cap S_2 \cap \ldots \cap S_{n-1}|$$

$$+ |S_n| - \sum_{i=1}^{n-1} (S_i \cap S_n) + \sum_{1 \le i < j \le n-1} |S_i \cap S_j \cap S_n| - \ldots + \qquad + \qquad + (-1)^{n-1+1+1} |S_1 \cap S_2 \cap \ldots \cap S_n|)$$

$$= \sum_{i=1}^{n} |S_i| - \sum_{1 \le i < j \le n} |S_i \cap S_j| + \sum_{k < i < j \le n} |S_i \cap S_j \cap S_k| - \ldots \qquad + (-1)^{n+1} |S_1 \cap S_2 \cap \ldots \cap S_n|$$

So we have proven the theorem by induction.

▨

Math263
p235.

Thm: The number of subsets of size k of a set S of size n, is $\binom{n}{k}$

*Notation $\binom{n}{k} = \frac{n!}{(n-k)!(k)!}$

Pf. Pick one element $x_i \in S$ (n choses)

- Pick one element $x_2 \neq x_1, x_2 \in S$ (n-1 choices)

...

Pick $x_k \in S$, x_k is not $= x_1, x_2, \dots$ (n-k+1 choses).

We pick every subset of size k exactly k! times namely one for every permutation of the elements in that subset.

$$\left.\frac{n(n-1)\ (n-k+1)}{k!} = \frac{n!}{k!(n-k)!}\right\}$$ the number of subsets of size k

For the hats problem

$$8! - \underbrace{\sum_{i=1}^{8}}7! + \sum_{1\le i<j\le 8} 6! - \sum_{1\le i<j<k\le 8} 5! + \dots$$

$$= \binom{8}{0}8! - \binom{8}{1}7! + \binom{8}{2}6! - \binom{8}{3}5! + \dots = \sum_{i=0}^{8}(-1)^i\binom{8}{i}(8-i)!$$

Defn: For a function $f: A \to A$, $a \in A$ is a fixed point of f if $f(a) = a$

A derangement is a permutation with no fixed point.

Thm: The number of derangements from A to A is $\sum_{i=0}^{n}(-1)^i\frac{n!}{i!}$ where $n = |A|$

Math 263

pz 36

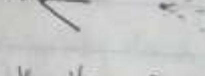
[MISSING].

<u>Defn</u>. $R(s,t)$ is the smallest number st all graphs on at least $R(s,t)$ vertices contains a clique of size s or a stable set of size t.) Ramsey number $R(3,3)=6$

<u>lemma</u>: $R(2,t)=t \quad \forall t \geq 2$) base case

<u>lemma</u>: $R(s,t) \leq R(s-1,t)+R(s,t-1) \quad \forall s \geq 3, t \geq t$.) induction find $R(s,t)$ in terms of 2 smaller ones

<u>Pf</u>. Let G be a graph with $R(s-1,t)+R(s,t-1)$ vertices.
Let $v \in V(G)$



for $R(3,3)$ look for a clique of size 2 or stable set of size 3)

v either has 3 neighbors or 3 non-neighbors

$V(G)$ has $R(s-1,t)+R(s,t-1)$

in this case, v has either $\underline{R(s-1,t)}$ neighbors or $\overset{|V(G)|-R(s-1,t)}{\underline{\quad}}$ non-neighbors.
If v has k neighbors, we look for a clique of size s-1 or a stable set of size t in $N(v)$
If we find a clique of size s-1, add "v" to that clique to obtain a clique of size s. If we find a stable set of size t, it is also a stable set of size t in G.

If v has $|V(G)|-R(s-1,t)=R(s,t-1)$ non neighbours, then we look for a clique of size s or a stable set of size T-1 in the non-neighbours of v.
If we find a clique of size s, then we found a clique of size s in G. Or if we find a stable set of size t-1, we add v to it and get a stable set of size t in G.
In all cases, we found a clique of size s or a stable set of size t in G

▨

Math263
p237

Thm: $R(s,t) \le 2^{s+t}$ $\forall s \ge 2, t \ge 2$

Pf By induction on s+t.

For $s+t=4$, $R(2,2)=2$ which is $\le 2^{2+2}$ (Base case).

Suppose theorem holds whenever $s+t=k-1$. If $s+t=k$

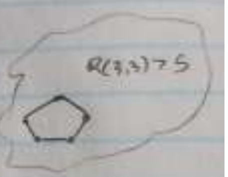
then $R(s,t) \le R(s-1,t)+R(s,t-1)$

$\le 2^{s+t-1}+2^{s+t-1}$ by induction

$= 2^{s+t}$

∎

(lame bound since $R(3,3)=6$ but thm caps it @ $2^{3+6}=2^6=64$).



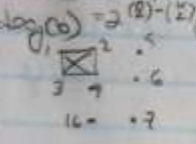
What about lower bound?

To prove a lower bound, we just need one graph (large) with no clique of size s and no stable set of size t.

[MISSING]. lower bound for Ramsey number $\frac{2^k}{2}$

lemma $\binom{n}{k}2^{-\binom{k}{2}}{}^{\times 2} < 1$ if $n<2^{k/2}$

$\log(co) = 2^{\binom{?}{2}-\binom{k}{2}}$ $\times 2$ double it



Pf $\binom{n}{k} = \frac{n!}{(n-k)!k!} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots(2)(1)}$ replace top terms by n (larger) / bottom terms by 2 (smaller)

$\le \frac{n\cdot n\cdot n\cdots n}{2\cdot 2\cdot 2\cdots 1} = 1$

$= \frac{n^k}{2^{k-1}}$

$< \frac{(2^{k/2})^k}{2^{k-1}} = 2^{\frac{k^2}{2}-k+1}$

Math 263 p738.

$\binom{n}{k}2^{-\binom{k}{2}} \times 2 < 2^{\frac{k^2}{2}-k+1-\binom{k}{2}+1}$

$= 2^{\frac{k^2}{2}-k+1-\frac{k^2}{2}+\frac{k}{2}+1}$

$= 2^{2-k/2} \le 1$ if $k \ge 4$

Linear Recurrences

[No need to know proofs]

$(L^2-L-I)\vec{F}=\vec{0}$

$(L-r)(L-s)\vec{F}=\vec{0}$

$F_n=cs^{n-1}+dr^{n-1}$

if it had the same root twice.

$(L-r)(L-r)\vec{a}=\vec{0}$

then the solution would have been $F_n=cr^{n-1}+dnr^{n-1}$

Discrete probability

Ω = Set of all possible outcomes. (Sample Space)

Probability Mass Function (PMF) $p:\Omega\to[0,1]$. w/ the property

$\sum_{e\in\Omega} p(e)=1$. (normalized)

Event. is a subset of Ω where $p[E]=\sum_{e\in E} p(e)$

Ex| Roll 3 die What is $P(10)$.

$\Omega=\{(i,j,k)\mid i|_1^6, j|_1^6, k|_1^6\}$ where $p(e)=\frac{1}{|\Omega|}$ $\forall e\in\Omega$

$E=\{(i,j,k)\mid i+j+k=10, (i,j,k)\in\Omega\}$

$P[E]=\sum_{e\in E}p(e)=\sum_{e\in E}\frac{1}{|\Omega|}=|E|\cdot\frac{1}{|\Omega|}$

$|E|=\underbrace{\binom{7+2}{2}}_{\text{regular}}-\underbrace{3\binom{1+2}{2}}_{\text{overflow because of restrictions: 712, 721, 811}}=27$

$i+j+k=10$

$(i-1)+(j-1)+(k-1)=10$

$\underbrace{(i-7)}_{\text{force the 1st die to 6}}+(j-1)+(k-1)=1$

Q2) 8 ppl in a restaurant and leave their hats @ the front. Then chose a hat when they leave. P. that they all have someone else's hat

$\Omega = \{$all permutations of $\{1, \dots, 8\}\}$.

$p(e) = \frac{1}{|\Omega|}$

$E = \{$all permutations of $\{1, 2, \dots, 8\}$ with no fixed points$\}$

$|\Omega| = 8!$

$|E| = \sum_{i=0}^{8} (-1)^i \frac{8!}{i!}$

So $p = \sum_{i=0}^{8} \frac{(-1)^i}{i!}$

3c) Suppose we throw an unusual 100 sided die, the prob. of getting i is $\frac{i}{\sum_{j=1}^{100} j}$. How likely are we to get a multiple of 2 or a multiple of 5.

$\Omega = \{1, 2, \dots, 100\}$

$p_i = \frac{i}{5050}$

let E_2 = Event to get a multiple of 2. $= \{2, 4, \dots, 100\}$

E_5 = Event to get a multiple of 5. $= \{5, 10, \dots, 100\}$

$E = E_2 \cup E_5$

$Pr[E] = Pr[E_2 \cup E_5] = \sum_{e \in E_2 \cup E_5} p(e) = \sum_{e \in E_2} p(e) + \sum_{e \in E_5} p(e) - \sum_{e \in E_2 \cap E_5} p(e)$

$= Pr[E_2] + Pr[E_5] - Pr[E_{10}]$.

$= \sum_{i=1}^{50} \frac{2i}{5050} + \sum_{i=1}^{20} \frac{5i}{5050} - \sum_{i=1}^{10} \frac{10i}{5050} = \frac{2 \times 50 \times 51}{2 \times 5050} + \frac{5 \times 20 \times 21}{2 \times 5050} - \frac{10 \times 10 \times 11}{2 \times 5050}$

Math263
pt 40

On a game show, there are 3 doors. Pick a door. Host opens an unchosen door that does not contain the prize. Do you want to change your pick. Should they switch?

$\Omega = \{\overset{\downarrow}{W}LL, \overset{\downarrow}{L}WL, \overset{\downarrow}{L}LW\}$

$p(e) = \frac{1}{3} \; \forall e \in \Omega$

Say we pick door #1.

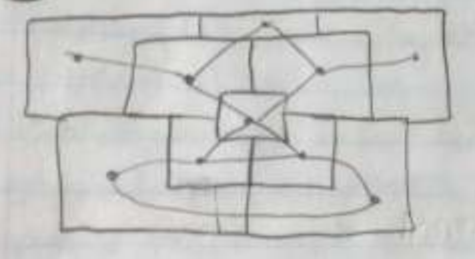
E_{switch} = win if we switch $\{\overset{\text{lose}}{WLL}, \overset{\text{win}}{LWL}, \overset{\text{win}}{LLW}\}$ So $p[E_{sw}] = 2/3$

E_{stay} = win if we stay. $\{\overset{W}{WLL}, \overset{L}{LWL}, \overset{L}{LLW}\}$ So $p[E_{stay}] = 1/3$

Planar graphs.

Fact: Every map can be colored w/ 4 colors s.t. countries that share boundaries have different colors. 

eg)



\# edges = # of boundaries
\# vertices = # of countries

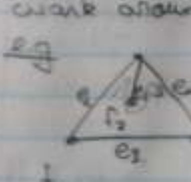
Defn: A graph is planar if it can be drawn in the plane s.t. no two edges cross.

Remark: There may be more than one way to draw a planar graph. eg) 

Defn: A planar graph with a drawing is called a plane graph. In a plane graph, the plane is split into regions which we call faces. (there is always a single ∞ large face called the outer face

Defn: $F(G)$ denotes the faces of a plane graph.
For $f \in F(G)$, $bd(f)$ is the boundary of f. It is a closed walk around f.

e.g.



$bd(f_1) = e_2, e_3, e_4$
$bd(f_2) = e_2, e_1, e_1, e_4, e_3$
} ordering is important but not the start point.

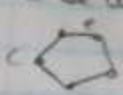
Notation: $|bd(f)|$ is the number of edges in the closed walk (some may be counted twice).

Famous Theorem: (Euler's formula) If G is a connected planar graph,
$|V(G)| + |F(G)| = |E(G)| + 2$

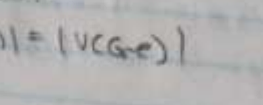
pf. (by minimum counter example).
Suppose thm is false. Then let G be a counter example minimizing the number of edges $|E(G)|$

Case 1 if G is a tree, then
$|E(G)| = |V(G)| - 1$
$F(G)| = 1$
$|E(G)| + 2 = |V(G)| - 1 + 2 = |V(G)| + |F(G)|$ ※

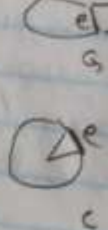
Case 2: G is not a tree, then G contains a cycle C. Let e be any edge in C. $G-e$ is connected.



$G-e$ is a plane graph. By minimality,
$|V(G-e)| + |F(G-e)| = |E(G-e)| + 2$.
Since we only removed an edge, $|V(G)| = |V(G-e)|$
By minimality
$|V(G-e)| + |F(G-e)| = |E(G-e)| + 2$
$|V(G)| = |V(G-e)|$
$|E(G)| = E(G-e)| + 1$
$|F(G)| = |F(G-e)| + ?$
Substitute
$|V(G)| + |F(G)| - 1 = |E(G)| \cancel{-1} + 2$



case: e is in bd of 2 faces.



You can't go from the inside to the outside w/o crossing

Math 263
pg 42.

Correlary: If G is a connected (simple) plane graph then $|E(G)| \le 3|V(G)| - 6$

pf. $\sum_{f \in F} |bd(f)| = 2|E|$

Every boundary has at least 3 edges  parallel ✱

$$2|E| = \sum_{f \in F} |bd(f)|$$
$$\ge \sum_{f \in F} 3 = 3|F| = 3(|E| - |V| + 2)$$ Euler

Rearrange $3|V| - 6 \ge |E|$

∎

Thm. Every Planar graph is 6-Colourable.

Rather than coloring the faces of a graph G, we will be coloring the vertices of the dual G*

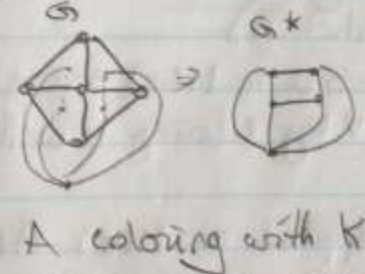
$V(G^*) = F(G)$

Each vertex ø f is incident to all edges in bd(f).

Equivalently, $E(G^*) = E(G)$ but an edge e now connects the 2 faces that contained it.

$F(G^*) = V(G)$

e.g



Dfn: A coloring with K colors (or K-coloring) of a graph G an assignment $c: V \to \{1, 2, \dots, K\}$ s.t. adjacent vertices are assigned different colors. i.e. $\forall e = (u,v) \in E$, $c(u) \neq c(v)$

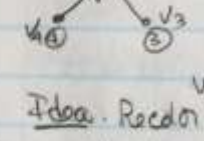
Defn: If there is a K-coloring of G, then we say that G is K-colorable

Math 263

PZ 43.

Pf by contradiction.
Suppose it is not the case, then $\exists$ a counter example which minimizes $|V(G)|$
$|V(G)|$ is non-empty
By corollary,
$6|V|-12 \geq 2|E| = \sum_{v\in V} deg(v)$ by Handshaking lemma.
So the average degree in G is < 6
Therefore, $\exists$ $v \in V$ with $deg(v) \leq 5$
By minimality, there is a 6-coloring "c" of G-v
We can extend c to v
Assign v a color that does not appear on $N(v)$. Such a color exists because there are 6 colors but at most 5 neighbors of v
∎

thm: Every planar graph is 5-colorable
pf: Suppose not. Let $G=(V,E)$ be a counter example minimizing $|V(G)|$
$\exists v \in V$ with $degree(v) \leq 5$.
By minimality, there exist a coloring on G-V (since V was the first vertex added that fucked shit up).
If $deg(v) \leq 4$ or $deg(v)=5$, and 2 vertices in $N(v)$ are given the same color, then extend c to V by coloring v with a color missing from $N(v)$
If $deg(v)=5$ and all 5 colors appear on $N(v)$ then



without loss of generality (swap till true), $N(v)=\{v_1,v_2,v_3,v_4,v_5\}$, $C(v_i)=i$ and they appear in cyclic order.

v with 1 and
Idea. Recolor v_1 with 3 then fix. Guaranteed no cycle but could stop with $v_3=1 \Rightarrow$ same color at v.

Math263
p7 44

Try again with v_2 if cycle, then G was not planar.

Let $G_{1,3}$ be the subgraph of G induced by the vertices colored 1 and 3.

if v_1 and v_3 are in different connected components of $G_{1,3}$, then swap colors 1 and 3 in the connected component containing v_1.

If v_1 and v_3 were in the same connected component, then let $G_{2,4}$ be the graph induced by the vertices colored 2,4. v_2 and v_4 are in different connected components. Swap colors 2 and 4 in the connected component containing v_2. Now $N(v)$ is missing a color, so simply pick that color to color v.

▨

Bipartite: No triangles -

Kuratowski's thm: A graph G is planar iff G does not contain a subdivision of K_5 or $K_{3,3}$.

$|V(K_{3,3})| = 6$

$|E(K_{3,3})| = \frac{18}{2} = 9$.

$2|V(K_{3,3})| - 4 = 8 < 9$. So not planar

tighter bound works for graph w/ no triangles. Bipartite can't have triangles

$|V(K_5)| = 5$

$|E(K_5)| = \binom{5}{2} = 10$.

if planar, $3|V(K_5)| - 6 = 9 < 10$ So K_5 can't be planar

K_4

Subdivision of K_4.

Math 263

p245.

Proposition: Statement that is either T/F (p, q, r)
Tautology: Prop. that is always true
Contradiction: " " False

$p \to q \Rightarrow \bar{p} \vee q$

p	q	p→q
1	1	1
1	0	0
0	1	1
0	0	1

ex) $p = $ "$\sqrt{2} > \frac{2}{3}$" & $q = $ "$(\sqrt{2})^4 > (\frac{3}{2})^4$"
Even though $\bar{p}$ & $\bar{q}$, $p \to q$

Inference Rules

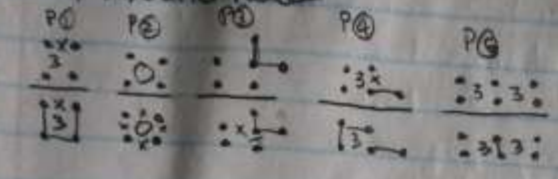
Name	Result	Name	Result	Name	Result
$\wedge e$	$\begin{array}{c} p \wedge q \\ \hline p \end{array}$	$\wedge i$	$\begin{array}{c} p \\ q \\ \hline p \wedge q \end{array}$	$\bar{i}$	$\begin{array}{c} p \to F \\ \hline \bar{p} \end{array}$
$\to e$	$\begin{array}{c} p \to q \\ p \\ \hline q \end{array}$	$\to i$	$\begin{array}{c} p \\ q \\ \hline p \to q \end{array}$	$\bar{e}$	$\begin{array}{c} p \\ \bar{p} \\ \hline F \end{array}$
$\vee e$	$\begin{array}{c} p \vee Q \\ p \to R \\ q \to R \\ \hline R \end{array}$	$\vee i$	$\begin{array}{c} p \\ \hline p \vee q \end{array}$	$\bar{\bar{e}}$	$\begin{array}{c} \bar{\bar{p}} \\ \hline p \end{array}$

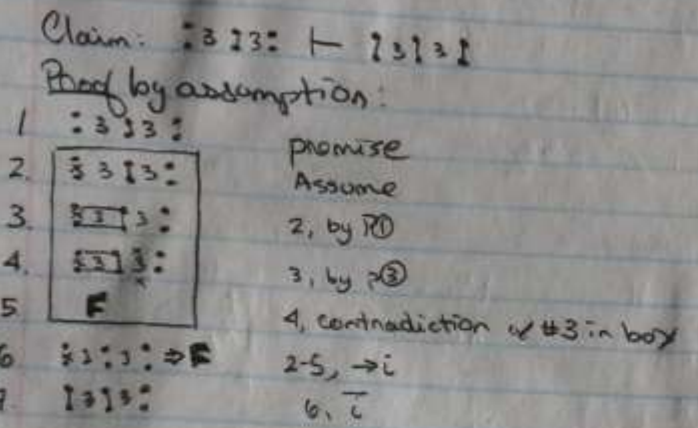
ex) 1. $p \to q$ promise
2. $\bar{q}$ promise
3. p assumption
4. q 1, 3, $\to e$
5. **F** 2, 4, $\bar{e}$
6. $p \to$ **F** 3-5, $\bar{i}$
7. $\bar{p}$ 6, $\bar{i}$

- In a proof, there should be no assumptions left @ the end, only promises
- Boxes = sub-proof

Math 263
p01

Ex of Inference rules




Claim: $\vdash$

Proof by assumption:

1 promise

2. Assume

3. 2, by P①

4. 3, by P③

5. F 4, contradiction w/ #3 in box

6. ⇒ F 2-5, →i

7. 6, $\overline{i}$

<u>Valid argument</u>: "$P_1, \dots, P_k \vdash Q$" is valid if there is a proof w/ promises $P_1, \dots, P_k$ & Conclusion Q (otherwise invalid)

<u>$\models$ proof by truth table</u>: for ∀ value of the variables which make $P_1, \dots, P_k$ true, Q is also made true.

ex $\overline{q}, p \to q \models \overline{p}$

p	q	$\overline{q}$	$p \to q$	$\overline{p}$
1	1	0	1	0
1	0	1	0	0
0	1	0		
0	0	1	1	1

LHS=T ⇒ RHS=T LHS RHS

*Only not allowed is 1,1→0

00
01
10 } don't imply anything since LHS≠T.

<u>Completness ($\models \vdash$)</u>: Can we prove every tautology?
$P_1, \dots, P_k \models Q$ then $P_1, \dots, P_k \vdash Q$ If you can prove w/ truth table, then you can prove w/ prop. logic.

Soundness ($\vdash \models$): Is everything we can prove a tautology?
$P_1, \dots, P_k \vdash Q$ then $P_1, \dots, P_k \models Q$ If you can prove w/ prop logic, then you can prove w/ prop. logic

Math263
p02

ex)

knights = always tell truth (true)
Natives = always lie (False)
Paul says: "If Quin is a native, then I am a knight"
Quinn says: "We are both different
mathematically:
$p \leftrightarrow (\bar{q} \to p)$
$q \leftrightarrow (\bar{p} \wedge q) \vee (p \wedge \bar{q})$

Truth table

Both premises need to be true

p	q	③ $\bar{q} \to p$	④ $(\bar{p} \wedge q) \vee (p \wedge \bar{q})$	$p \leftrightarrow$ ③	$q \leftrightarrow$ ④
1	1	1	0	1	0
1	0	1	1	1	0
0	1	1	1	0	1
0	0	0	0	1	1

← only solution

Methods of Proof

Direct Proof: Assume P, get Q, proved that $P \to Q$

Indirect Proof: Assume $\bar{Q}$, get $\bar{P}$, proved that $P \to Q$.

ex) Dfn: n odd $\Rightarrow n = 2k+1$
n even $\Rightarrow n = 2k$.
} an integer n is either even or odd.

Claim: if $3x+2$ is odd, then x is odd

Proof:
Suppose x even, $x = 2k$.
Then $3x+2 = 3(2k)+2 = 2(3k+1) \Rightarrow$ multiple of 2 is even
Therefore $3x+2$ is even

$P \to Q$ proved by $\bar{Q} \vdash \bar{P}$

Math 26
p03.

Proof by Contradiction::

- Prove $P \to Q$ by $(\overline{P \to Q}) \to \mathbb{F}$
- Prove P by $\overline{P} \to \mathbb{F}$

ex Claim there are ∞ number of primes.

Proof. Suppose finite # of primes.

let $n = p_1 p_2 \dots p_k = \prod_{i=1}^{k} p_i$

* $\forall p_i$ devides n, so p_i doesn't devide $n+1$

* $n+1 > \forall p_i$, therefore was not in initial set

$\Rightarrow$ Contradiction, so ∞ # of primes.

Predicate: Function from a domain or universe of discourse that is either {T, F} they are allowed to take more than one input

ex AreEqual(x, y) = "x = y" is a predicate.

Contradiction: ✱

Universal Quantifyer ($\forall$): $\forall x P(x) \Rightarrow$ for every x in the domain, $P(x)$ is true

Existential Quantifyer ($\exists$). $\exists x P(x) \Rightarrow$ There exists an x in the domain for which $P(x)$ is true.

ex $\forall x \exists y \, p(x,y)$ where $p(x,y) =$ "$x+y=0$" is $\mathbb{T}$

$\exists x \forall y \, p(x,y)$ where $p(x,y) =$ "$x+y=0$" is $\mathbb{F}$

Negating

$\overline{\forall x P(x)} \leftrightarrow \exists x \overline{P(x)}$

$\overline{\exists x \, p(x)} \leftrightarrow \forall x \overline{P(x)}$

$\overline{\forall x \exists y \, P(x,y)} \leftrightarrow \exists x \left[\overline{\exists y \, p(x,y)}\right] \leftrightarrow \exists x \forall y \overline{p(x,y)}$

Propositional Equivalences: P & Q are logically equivalent if $P \leftrightarrow Q$ is a tautology. $P \equiv Q$

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
$p \vee (p \wedge q) \equiv p$
$p \wedge (p \vee q) \equiv p$
$p \rightarrow q \equiv \bar{p} \vee q$

Graph Theory

- **Walk**: Alternating sequence of vertecis & edges in which consecutive elements are adjacent
- **Trail**: Walk where no edge is repeated
- **Path**: Trail where no vertex is repeated.

- **Simple graph**: No loops $(xx \in E, x \in V)$ & No multiedges $(x \bigcirc y)$

Observations

- Concatenation of ∀ 2 walks is a walk
- If 2 trails are edge disjoint, then their concatenation is also a trail

Handshaking lemma

$\Sigma \underbrace{d(v)}_{\text{degree}} = 2|E(G)|$

Since 1 edge touches 2 vertices,
$\Sigma d(v) + 1E$
$\Sigma d(v) + 2$ degrees

Matching: M in a graph E is a set of edges s.t. every vertex has at most one edge

M-augmenting path: Endpoints are not in edges of M, and edges of P alternate between being in/out of M

Solution

① Start w/ any matching M.

② Find an M-anything path P

③ Switch on P (in/out M).

④ Repeat until no more augmenting paths.

Δ = Simetric difference. $A \Delta B$ = either in A or B but not in both

Proof of correctness

Suppose M, N are matchings w/ no M-augmenting paths, but $|N| > |M|$.

Consider $M \Delta N$ edges in M or N, but not in both. Each vertex of E is in ≤ 2 edges of $M \Delta N \Rightarrow M \Delta N$ consists of paths & cycles.

- Cycles are even 
- at least 1 path starts & ends w/ an edge of N since $|M| > |M|$ 

  this is M-augmenting path

⇒ Assumed none but found one = ⚡

Eulerian trail (path) trail which contains every edge exactly once

Eulerian circuit Eulerian trail where start/end vertex are the same

Lemmas: If a Graph contains an eulerian circuit

- ∀ vertex of G has even degree
- The graph is connected
- Graph has no vertex of degree 0.

} sufficient conditions to conclude that path contains Eulerian circuit.

"Pf" Start anywhere, keep moving until you return to start

Math 263
p06.

G has an <u>Eulerian trail</u> iff
- G is connected
- G has at most 2 odd degree vertices.

<u>lemma</u>: if G has an Eulerian trail, then there is an edge we can add s.t. G has an Eulerian circuit.

<u>Hamiltonian path</u>. in a <u>graph</u>, a path which contains every vertex exactly once

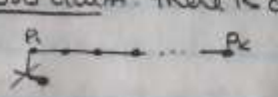
<u>Hamiltonian cycle</u>. cycle that contains every vertices.
- Every vertex has $\deg \geq 2$
- G is 2-connected

} not sufficient conditions.

<u>Diarc's Thm</u>: If a simple graph w/ 3+ vertices has $\deg(v) \geq \left|\frac{V(G)}{2}\right|$ for $\forall v \in V(G)$, then G has a Hamiltonian <u>cycle</u>.

<u>Pf</u>. Let $p_1, \ldots, p_k$ be the longest path in G

<u>Sub claim</u>: there is a cycle of lenght k in G



p_1 is only adjacent to vertices in path, otherwise, there is a longer path $\Rightarrow$ Same for P_k

Since $\deg(p_1) \geq n/2$, and so is $\deg(p_k) \geq \frac{n}{2}$, there exists i st p_1 adj(i) and p_k adj$(i \pm 1)$.

$\Rightarrow$ Worst case, $n = k$ (all vertices in path).

Hypothesis is true for some i

$|S_1| = |\{j \mid p_1 p_j \in E(G)\}| = \deg(p_1) \geq n/2$

$|S_2| = |\{j+1 \mid p_k p_j \in E(G)\}| = \deg(p_k) \geq n/2$

is there a number i st $i \in \{S_1 \wedge S_2\}$? if so, p_1 is adjacent to p_i and p_k is adjacent to p_{i-1}. Otherwise, $S_1 \wedge S_2 = \{\}$

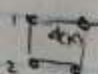
Math 263
p07.

Both S_1 and S_2 only contain numbers between 2 and k, and k is at most $|V(G)|$
if we mark "x" as cannot be neighbors of p_k, $\geq \frac{n}{2}$ marked,
So $\leq \frac{n}{2}$ left to be adjacent to p_k

That is, at most n-1 numbers, and each set has at least n/2 numbers in it ⇒ So the sets have to intersect.

Fact: we have a cycle of length k in G

Let $c_1, \dots, c_k$ be a cycle of length k. then there are no edges between the vertices in C and those not in C, otherwise, there exists a path longer than k

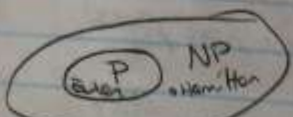


(would make 3-2-1-4-5 longer than 1-k)

The cycle c has @ least $\frac{n}{2}+1$ vertices, so there are at most $\frac{n}{2}-1$ vertizes not in C. Those vertizes only have neighbors not in C, but out of $(\frac{n}{2}-1)$, there are not ennough vertices to satisfy $deg(v) \geq \frac{n}{2}$ (assumption on degree of $v \in V(G)$)
※ ■

NP: Non deterministic polynomial
A problem is NP-complete if it is in NP & it is NP hard
⇒ Polynomial time = $O(n^k)$

Deciding if a graph has a Hamiltonian cycle is NP complete



Math 363
p08

Hypercubes: Label vertices by a n-bit number, 1bit/dimension. 2 vertices are adjacent if they differ by 1 bit.

Theorem: for $n \geq 2$, Q_n has a Hamiltonian cycle.

Pf: there is a sequence containing all binary strings s.t. all consecutive strings differ by only one bit => called Grey Codes.

Greycodes

① Write down (n-1) sequence.

② Copy, reverse, paste

③ add 0/1 on both sub/sequences.

Ex n=3

write n=2	copy	reverse	paste	slap 0/1
00	00	11	00	000
01	01	10	01	001
10	10	01	10	010
11	11	00	11	011
			11	111
			10	110
			01	101
			00	100

TREE: Connected graph with no cycles.

Rooted tree: Obtained from a tree T and a root $r \in V(T)$ by directing all edges towards the root.

Lemma: A tree w/ 2+ vertices has a vertex of degree 1

Pf: Suppose false.

- T has no vertex of degree ∅ because connected
- T has no vertex of degree 1 because false lemma.
- All $v \in V(T)$ have $\deg(v) \geq 2$

Start @ any vertex and walk using only 'unused' edges. either

① We got stuck => impossible since all $\deg \geq 2$.

② We revisit a vertex = cycle = Not a tree.

$\divideontimes$ ☒

<u>Lemma</u>: A tree w/ n vertices has exactly $n-1$ edges.

Pf: Base case: if $n=1$, • has 0 edges.

Induction Step:

Let T be a tree on n vertices. By the previous lemma, it has at least one vertex v w/ $\deg(v)=1$

Claim that $G-v$ is still a tree

- Deleting a vertex and edges around it does not create a cycle
- $G-v$ is still connected. For any vertex pair $u,w \in G$, the path that connected them before deleting v is still there

So by removing a vertex of $\deg=1$, we remove exactly one vertex & one edge, and the remaining graph is still a tree

<u>Lemma</u>: If $T=(V,E)$ is a tree and $e \notin E$, then $T=(V, E\cup\{e\})$ contains exactly one cycle C which contains e.

Pf Let $e=(u,v)$. Since T is connected, $\exists$ path P_1 between

u & v in T. Sine e connects both 'tips' of the path, it becomes a cycle.

Claim that there are no other path P_2 from u to v in T. Otherwise, P_2 concatenated with P_1 is a cycle. If there is another cycle in $T+e$, then removing e from that cycle gives a second path P_2 between u and v # ▨

<u>Minimum Spanning Tree</u>

Input: Connected graph $G=(V,E)$ and weights w_e for all $e \in E$

Output: Connected subgraph (V,F) of G minimizing $\sum_{e\in F} w_e$.

Math 263

p10

Kruskal's Algorithm to solve MST.

① Start w/ empty set $F = \{\}$.
② Sort all edges in ascending weights
③ Iterate over sorted edges with
    If $(V, F \cup \{e\})$ has no cycles, $F \leftarrow F \cup \{e\}$.
④ Return F.

Pf of correctness
Need to show
① F is connected
② F has no cycles.
③ F has minimum weight

Pf1: Let $u, v \in V$. Since G is connected, $\exists$ a path P from u to v. Replace each edge of P not in F by the edge of the cycle in $F \cup \{e\}$

Pf2: Starting w/ $F = \{\}$ has no cycles. Given that it has no cycles in some iteration, F has no cycles in the next because of selection criteria.

Pf3: Start w/ $F = \{\}$. Since there are no double edges, we know that we have to stack at least 2 edges before we need to check for cycles. So weight of F is minimal before we get to make a choice. Then we pick the low edge, unless it creates a cycle.

Thm Kruskal's algorithm returns a MST.

Pf: Need to show
1- F is connected
2- F has no cycles
3- F has minimum wheight

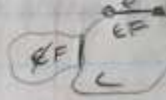
Pf1: Let $u, v \in V$ Since G is connected, $\exists$ path P from u to v. Replace each edge of P not in F by the edges of the cycle in $F \cup \{e\}$

Pf2: At the beginning, $F = \{\}$ has no cycles. Given that it has no cycles in some iteration, it has no cycles in the next iteration because of condition imposed on adding an edge to F.

Pf3: Let F^* be a MST which minimizes $|F \cap F^*|$ Suppose $F \neq F^*$.

set minus
(C)-F ≡ ?s then
(c)}-{F}

Look at the first time we add an edge e to F which is not in F^*.



$F^* \cup \{e\}$ Contains a unique cycle C (and e is in it)
Let f be an edge in $E(C) \setminus F$ (edges in cycle not in F) with maximum weight. We claim $w_f \geq w_e$. If not, we could have picked f instead of e in this iteration. Only reason we would not pick f is because it would create a cycle.

Redone here.

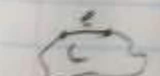
PP3: the output of Kruskal's algorithm is minimum

Let F be the output. Let F^* be a MST which ~~minimizes~~ maximizes $|F \cap F^*|$. If $F = F^*$, we have proven that the output is minimum. Otherwise, look at the first iteration in which we pick an edge $e \notin F^*$.

Math 263
pt 15

This means we went from F_i to $F_i \cup \{e\}$ in that iteration

By lemma $(V, F^* \cup \{e\})$ has a unique cycle C.



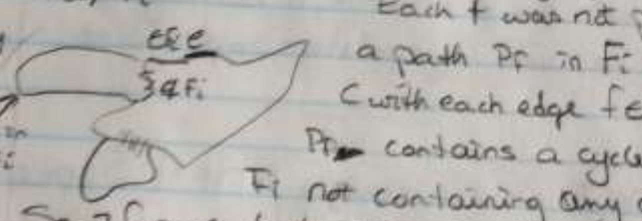
Since $F_i \cup \{e\}$ contains no cycles, there is an edge f in C that is not in $F_i \cup \{e\}$

We claim

~~Pf by contradiction. If not, $w_f < w_e$ we could have picked f instead of e.~~

We claim $\exists f$ s.t. $w_f \geq w_e$ If not, $w_f < w_e \ \forall f \in E(C) \setminus F_i$
Why did we pick e instead of any of the edges in $E(C) \setminus F_i$

all edges of lower weight than e would have created a cycle that's why they was not picked in the first place



Each f was not picked because there is a path P_f in F_i between its endpoints. C with each edge $f \in E(C) \setminus F_i$ replaced by P_f contains a cycle Contradiction to F_i not containing any cycles. ▨

So $\exists f$ in C but not in F_i with $w_f \geq w_e$
$(V, F \cup \{e\} \setminus \{f\})$ contains no cycles since the unique cycle C we got from adding e. ⇒ C is the unique cycle in $(V, F^* \cup \{e\})$
$(V, F^* \cup \{e\} \setminus \{f\})$ has weight less or equal to F^* and it contains more edges of F than

$\not\longrightarrow$

What if we wanted to pick (minimum weight) edges s.t. the resulting graph (V, F) is 2-connected (instead of just 1-connected)
This is NP-hard

Math263
pg 16

De Morgan's proofs.

$\overline{r \vee g} \rightarrow \bar{r} \wedge \bar{g}$

1. $\overline{r \vee g}$ premis
2. r assume
3 $r \vee g$ $\vee i$
4 F 1, cat.
5 $\bar{r}$
6. g as
7 $g \vee r$ $\vee i$
8 F
9. $\bar{g}$
10. $\bar{r} \wedge \bar{g}$ 5, 9.

$\bar{r} \wedge \bar{g} \rightarrow \overline{r \vee g}$

1 $\bar{r} \wedge \bar{g}$ premis.
2. $r \vee g$ assume
3 r as.
4 $\bar{r}$ 1.
5 F
6. $r \rightarrow F$ 3-5.
7. g as.
8.
9. [illegible]
10 $g \rightarrow F$
11. F 2, 6, 10
12. $r \vee g \rightarrow F$
13. $\overline{r \vee g}$

$\overline{r \wedge g} \rightarrow \bar{r} \vee \bar{g}$

1. $\overline{r \wedge g}$ premis
2. $\overline{\bar{r} \vee \bar{g}}$ assume
3. $\bar{r}$ ass.
4 $\bar{r} \vee \bar{g}$ $\vee i$
5. F
6. r 3-5.
7. $\bar{g}$ as.
8. $g \vee \bar{r}$ $\vee i$
9. F
10. g 7-9
11. $r \wedge g$ 6, 10.
12 F 1, 11
13 $\overline{\bar{r} \vee \bar{g}} \rightarrow F$
14 $\bar{r} \vee \bar{g}$

$\bar{r} \vee \bar{g} \rightarrow \overline{r \wedge g}$

1. $\bar{r} \vee \bar{g}$ premis.
2. $r \wedge g$ assume
3. $\bar{r}$ as.
4 r 2.
5 F
6. $\bar{r} \rightarrow F$ 3-5.
7. $\bar{g}$ as.
8 g 2
9 F
10. $\bar{g} \rightarrow F$
11. F 1, 6, 10
13. $r \wedge g \rightarrow F$ 2-11
14. $\overline{r \wedge g}$