Lecture 8:

A **binary relation** R from A to B consists of:

· Set A (Domain) · Set B (Co Domain)

· Subset graph(R) $\subseteq A \times B$

Can visualize w/ **relation diagram**.

The relation R^{-1} (inverse R) has domain B, codomain A and $graph(R^{-1}) := \{(b,a) \in B \times A \mid aRb\}$

$\rightarrow bR^{-1}a = aRb$

Functions: $aRb \wedge aRb' \Rightarrow b = b'$

A func from A to B has for all $a \in A$, the set $\{b \in B \mid aFb\}$ has ≤ 1 elt.

↳ ie. only $\leq$ one arrow coming out of any $a \in A$.

total if: each $a \in A$ has ≥ 1 outgoing arrow

injective (1-to-1) if: each $b \in B$ has ≤ 1 incoming arrow

Surjective if: each $b \in B$ has $\geq$ incoming arrow

bijective if: each $a \in A$ has 1 outgoing and each $b \in B$ has 1 incoming

Thm. If f is a total function, then f is bijective iFF it is injective and surjective.

Thm. Bijection from A to B $\Rightarrow |A| = |B|$

Thm. $\forall n \geq 0, (G^n \circ G) = G^{n+1}$ Thm. $\forall n \geq 1, A_{G^n} = (A_G)^n$

Lemma: $\forall n \geq 0, A_{G^n \circ G} = A_{G^n} * A_G = A_{G^{n+1}}$

Connectivity: u connected to v if $\exists$ walk from u to v

Def. $G = (V, E)$, G^* def by $E' = \{(u,v) \mid u \text{ connected to } v\}$

Thm. $G^* = (G^{\leq})^{n-1}$ where $G^{\leq}$ is G w/ 1's all along main diagonal

$\rightarrow G^{\leq} := (V \in V\ \{(v,v) \mid v \in V\})$

Relation Composition:

If $R: A \rightarrow B$ and $T: B \rightarrow C$, then the composition of R & T, $T \circ R$ is a relation from A to C

⭑ $T \circ R = T(R(x))$ ... R happens 1st.

Range (R) is all the elts w/ arrows coming IN.

Directed Graphs:

$G = (V, E)$ has a set V of vertices and set $E \subseteq V \times V$ (edges)

· G is a binary relation on V w/ E = graph(G)

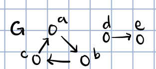
Walks: A sequence $V_1, \dots, V_K \in V$ s.t. $\forall i = 1, \dots, k, (V_i, V_{i+1}) \in E$

↳ K vertices are traversed in K-1 edges

Path: A walk in which all $V_1, \dots, V_K$ are distinct

Adjacency Matrices:

Put 1 in (i,j) iFF $(i,j) \in E$ to get adjacency matrix, A_G



$$\begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 1 & 0 & 0 & 0 \\ b & 0 & 0 & 1 & 0 & 0 \\ c & 1 & 0 & 0 & 0 & 0 \\ d & 0 & 0 & 0 & 0 & 1 \\ e & 0 & 0 & 0 & 0 & 0 \end{array} \quad A_G$$

Given $G = (V, E)$ and $n \in \mathbb{N}$,

$G^n = (V, E')$, $E' = \{(u,v) \mid \exists \text{ length } n \text{ walk from } u \text{ to } v.\}$

$\rightarrow u G^n v$ iFF $\exists$ length n walk from u to v.

Lecture 9:

Directed Acyclic Graphs (DAGs):

· A directed graph with NO cycles.

Topological Sort of a DAG is list of nodes such that each node, v, appears earlier in the list than EVERY other node reachable from v.

↳ Find by **Greedy Algorithm**:

Repeatedly pick minimal elts, remove from graph & add to list.

· **Minimal Elements**: Elements with no "in-arrows"

⭑ After processing a minimal elt, you might uncover new minimal elts

Nodes u,v are **comparable** in DAG if u can reach v OR v can reach u.

A **chain** is a set of nodes s.t. any pair is comparable

In a chain, a node reachable from all others is the **maximum element** of the chain

An **antichain** is a set of nodes s.t. no two nodes are comparable

↳ You can process all minimal nodes in an antichain at the same time!

rounds required $\geq$ length of any chain

⭑ Longest chain is called **critical path** $\Rightarrow$ # rounds = length of critical path.

rounds $\leq$ length of largest chain/critical path.

Dilworth's Theorem: Every DAG on n nodes has either a chain of size > t or antichain of size n/t

Linear order: all pairs of diff. elts are comparable. Can tell which comes before/after ALWAYS

Binary Relation R on A is:

Transitive iFF $\forall a,b,c \in A, (aRb\ \&\ bRc) \Rightarrow (aRc)$

Reflexive iFF $\forall a \in A$, aRa

Irreflexive iFF $\forall a \in A$, $\neg(aRa)$

Symmetric iFF $\forall a,b \in A$, $aRb \Rightarrow bRa$

Asymmetric iFF $\forall a,b \in A$, $aRb \Rightarrow \neg(bRa)$

Antisymmetric iFF $\forall a \neq b \in A$, $aRb \Rightarrow \neg(bRa)$

R is an **equivalence relation** if it is reflexive, symmetric, and transitive

Strict Partial Order is transitive & irreflexive.

↳ Thm. R is S.P.O. iFF transitive and asymmetric.

Weak Partial Order is transitive, reflexive, and antisymmetric.

Lecture 13: Asymptotics

Notation	Test / Rule	
$f \sim g$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$	
$\leq$ $f \in O(g)$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} \in \mathbb{R}$	$\exists c \exists M. \forall x > M: \lvert f(x)\rvert \leq c\,g(x)$
$<$ $f \in o(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} = 0$	
$\geq$ $f \in \Omega(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} \in (0, \infty]$	when $g \in O(f)$
$>$ $f \in \omega(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} = \infty$	when $g \in o(f)$
$=$ $f \in \Theta(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} \in (0, \infty)$	$f \in O(g)$ and $f \in \Omega(g)$

$f \in \omega(g) \Rightarrow f \notin O(g)$

$f \in o(g) \Rightarrow f \notin \Omega(g)$

Lecture 11

For simple graphs, u & v connected if there is a path between them

As a binary relation, connectivity is:

Reflexive: u & u always connected

Transitive: u connect v & v connect w $\Rightarrow$ u connect w

Symmetric: If u connected to v, then v connected to u.

In simple graphs, connectivity is an equivalence relation.

G is **Tree**:

· Connected and acyclic

· Connected & every edge is cut edge

· Acyclic but adding edge creates cycle

· Every pair of vertices connected by unique path

· Connected, exactly n-1 edges

· Acyclic, exactly n-1 edges.

A **cut edge** is an edge s.t. removing it disconnects two vertices.

A graph is connected if it has 1 **connected component**

↳ Every vertex is connected to every other vertex in a connected component

Bipartite Graph: Graphs w/ two sides

$G = (V, E)$ is bipartite if there are sets L & R s.t.

1). $L \cup R = V$

2). $L \cap R = \emptyset$

3). $E \subseteq \{\{l, r\} \mid l \in L, r \in R\}$

All edges have one endpoint in L and one endpoint in R.

G is bipartite iFF G is 2 colorable.

G is bipartite iFF every cycle has even length.

Trees are bipartite graphs

Bipartite Matching:

set of |L| edges s.t. every $l \in L$ appears once and every $r \in R$ appears ≤ 1 time.

↳ Can have more edges in R than L.

$|L| = |R|$ is perfect matching

Hall's Theorem:

There is matching iFF there are no bottlenecks

$(\forall S \subseteq L, |N(S)| \geq |S|)$

where $N(S) = \{r \in R \mid \exists l \in S \text{ s.t } \{l, r\} \in E\}$

Lecture 10:

$G = (V, E)$ consists of:
Set V of vertices
Subset E of $\{\{u,v\} \mid u,v \in V,\ u \neq v\}$

Handshaking Lemma: $\sum_{v \in V} \deg(v) = 2|E|$

u is **adjacent** to v iff $\{u,v\} \in E$
edge $\{u,v\}$ is **incident** to u & v... u & v are endpoints of edge $\{u,v\}$
The **degree** of $v \in V$, $\deg(v)$, is the # of incident edges to that node
For $n \geq 1$, the max # edges in an n-node graph is $\frac{n(n-1)}{2}$

Graph Isomorphisms

$G = (V, E)$ and $G' = (V', E')$ are isomorphic if
$\exists$ bijection $f: V \to V'$ that "preserves edges"
$\forall u,v \in V$, $\{u,v\} \in E$ iff $\{f(u), f(v)\} \in E$
Same graph, diff layouts ok
Same graph, diff node labels ok

P is preserved by isomorphism if for all graphs G, H:
($P(G)$ True $\wedge$ G is isomorphic to H) $\Rightarrow$ $P(H)$ True.
Proving Isomorphism: Simply give one example isomorphism
Proving Non-Isomorphism: check #vertices, $\deg(v)\ \forall v$, #edges.
Find P preserved by isomorphism, then show it doesn't hold for G & H.

Graph Coloring:

Proper Coloring, $c: V \to \{1, \ldots, k\}$ s.t. $\forall \{u,v\} \in E$, $c(u) \neq c(v)$
Chromatic Number $\chi(G)$ = min # colors needed for proper coloring
* Graph containing a triangle has min 3 colors
- Showing a k-coloring shows $\chi(G) \leq k$
- Show then that no $(k-1)$-coloring works

$\Rightarrow \chi(G) =$ exactly k.

4-Color Thm: Every Planar Graph is 4-colorable
Thm. If every vertex in G has deg. at most d, then $\chi(G) \leq d+1$

C_n is cycle on n vertices:
$\chi(C_{2k}) = 2$, $\chi(C_{2k-1}) = 3$
K_n is clique on n vertices (Max # edges):
$\chi(K_n) = n$

Lecture 14: Recurrences

Counting # of moves an algorithm takes:
- Let $T(n)$ be # moves for n elements.
- Find # moves for every step in the alg
  (May be written as $T(\text{something})$).
- Combine to find $T(n)$, the open form solution.

Find closed form of $T(n)$ by:

Guess and Verify
Plug in small values of n
Guess pattern
Prove guess by induction

Plug & Chug (Expansion, Iteration, Brute Force)
- Plug in expansions of recursive terms where possible. Simplify
- Repeat until you can guess a pattern
- Prove guess by induction

Guess may still have recursive term. Find params s.t. that $T(\text{something})$ terms disappears. See lec ex.

Divide and Conquer Recurrence:

A D&C recurrence given by
$$T(x) = g(x) + \sum_{i=1}^{k} a_i T(b_i x) \text{ for } x > x_0 \text{ that satisfies:}$$
1). For all i, $a_i > 0$ and $0 < b_i < 1$
2). g is differentiable and polynomially bounded
3). x_0 is "big enough"

Thm Akra-Bazzi
If $T(n)$ has D&C recurrence, and
p s.t. $\sum_{i=1}^{k} a_i b_i^p = 1$ Increasing p decreases sum bc $b < 1$ *
Then, $T(n) \in \Theta\left(n^p + n^p \int_1^n \frac{g(x)}{x^{p+1}}\,dx\right)$

When p is not easy to find: find what p must be less than s.t. $\sum_{i=1}^{k} a_i b_i^p < 1$
Then, execute Akra-Bazzi and find $T(n)$ with p still in exponents
Then, find dominating term given the bound on p we found.

Thm. If $g(x) \in \Theta(x^t)$ and $t \geq 0$ and $\sum_{i=1}^{k} a_i b_i^t < 1$
Then, $T(x) \in \Theta(g(x))$

Master Theorem (Simpler cases of Akra-Bazzi for $k=1$) Here, $p = -\log_{b_1}(a_1)$
Case 1: If $g(x) \in O(x^c)$ $c < p$
then $T(x) \in \Theta(x^p)$
Case 2: If $g(x) \in \Theta(x^p)$
then $T(x) \in \Theta(x^p \log x)$
Case 3: If $g(x) \in \Omega(x^c)$ $c > p$
then $T(x) \in \Theta(g(x))$

Some Common Sums:

For $|x| < 1$, $\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$. Diverges if $|x| \geq 1$

$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ Harmonic Series $H_n = \sum_{i=1}^{n} 1/i$

$\sum k^2 = \frac{n(n+1)(2n+1)}{6}$ $\sum_{i=1}^{\infty} i x^{i-1} = \frac{1}{(1-x)^2}$

$\sum k^3 = \frac{n^2(n+1)^2}{4}$ $\sum_{i=0}^{n} a r^i = \frac{a r^{n+1} - a}{r-1}$

Apply log to a product turns it into a sum:
$\ln(n!) = \ln(1) + \ln(2) + \ldots + \ln(n) = \sum_{i=1}^{n} \ln(i)$.

Lecture 12: Evaluating Sums

For a summation, a **closed-form** solution is an algebraic equation with the same value as the summation

Perturbation Method
- Combine a sum and a perturbation of that sum to get something useful.

ex. $S = \sum_{i=0}^{n-1} x^i$
$S = 1 + x + x^2 + \ldots + x^{n-1}$
$xS = x + x^2 + \ldots + x^{n-1} + x^n$
$\Rightarrow S - xS = 1 - x^n$ $S = \frac{1-x^n}{1-x}$

Smart Grouping:
Identify groupings to make your life easier!
ex. $S = \sum_{i=1}^{n} i$
$S = 1 + 2 + \ldots + n$ AND $S = n + n-1 + \ldots + 1$
so $2S = n(n+1)$ and $S = \frac{n(n+1)}{2}$

Derivative Method: Use derivative of known closed forms to find new sums

ex. $S = \sum_{i=0}^{n} i \cdot x^i$ Notice S looks like $\sum_{i=0}^{n} i x^{i-1} = \frac{d}{dx}\left(\sum_{i=0}^{n} x^i\right)$, where we know $\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$

$\rightarrow \sum_{i=0}^{n} i x^{i-1} = \frac{1-(n+1)x^n + n x^{n+1}}{(1-x)^2}$ ∴ multiply both sides by x to get what we want!

Approximate by **Integration Bounds** (when no closed form)

if f is positive & increasing: $f(1) + \int_1^n f(x)\,dx \leq \sum_{i=1}^{n} f(i) \leq f(n) + \int_1^n f(x)\,dx$ Flipped if decreasing

Lecture 9:

Directed Acyclic Graphs (DAGs):
- A directed graph with NO cycles.
  - ex. precedence relations (getting dressed)

Topological Sort of a DAG is list of nodes such that each node, v, appears earlier in the list than EVERY other node reachable from v.
↳ Find by "Greedy Algorithm":
Repeatedly pick minimal elts, remove from graph & add to list.

- Minimal Elements: Elements with no "in-arrows"
  - ★ After processing a minimal elt, you might uncover new minimal elts.

Nodes u,v are comparable in DAG if u can reach v OR v can reach u.
A chain is a set of nodes s.t. any pair is comparable
In a chain, a node reachable from all others is the maximum element of the chain

An antichain is a set of nodes s.t. no two nodes are comparable
↳ You can process all minimal nodes in an antichain at the same time!

rounds required ≥ length of any chain
★ Longest chain is called critical path ⇒ # rounds = length of critical path.
rounds ≤ length of largest chain/critical path.

Dilworth's Theorem: Every DAG on n nodes has either a chain of size $>t$ or antichain of size n/t

Linear order: all pairs of diff. elts are comparable. Can tell which comes before/after ALWAYS

Binary Relation R on A is:

Transitive iff $\forall a,b,c \in A, (aRb \text{ \& } bRc) \Rightarrow (aRc)$

Reflexive iff $\forall a \in A, aRa$

Irreflexive iff $\forall a \in A, \neg(aRa)$

Symmetric iff $\forall a,b \in A, aRb \Rightarrow bRa$

Asymmetric iff $\forall a,b \in A, aRb \Rightarrow \neg(bRa)$

Antisymmetric iff $\forall a \neq b \in A, aRb \Rightarrow \neg(bRa)$

Strict Partial Order is transitive & irreflexive.

↳ Thm. R is S.P.O. iff transitive and asymmetric.

Weak Partial Order is transitive, reflexive, and antisymmetric.

R is an equivalence relation if it is reflexive, symmetric, and transitive

Lecture 10:

$G = (V, E)$ consists of:

- set V of vertices
- subset E of $\{\{u,v\} \mid u,v \in V, u \neq v\}$

Handshaking Lemma: $\sum_{v \in V} \deg(v) = 2|E|$

u is adjacent to v iff $\{u,v\} \in E$

edge $\{u,v\}$ is incident to u & v... u & v are endpoints of edge $\{u,v\}$

The degree of $v \in V$, $\deg(v)$, is the # of incident edges to that node

For $n \geq 1$, the max # edges in an n-node graph is $\frac{n(n-1)}{2}$

Graph Isomorphisms

$G = (V,E)$ and $G' = (V', E')$ are isomorphic if

$\exists$ bijection $f: V \to V'$ that "preserves edges"

$\forall u,v \in V, \{u,v\} \in E$ iff $\{f(u), f(v)\} \in E$

Same graph, diff layouts ok

Same graph, diff node labels ok

P is preserved by isomorphism if for all graphs G, H:

$(P(G) \text{ True} \wedge G \text{ is isomorphic to } H) \Rightarrow P(H) \text{ True}$.

Proving Isomorphism: Simply give one example isomorphism

Proving Non-Isomorphism: check #vertices, $\deg(v)\ \forall v$, #edges.

Find P preserved by isomorphism, then show it doesn't hold for G & H.

Graph Coloring:

Proper Coloring, $c: V \to \{1, \dots, k\}$ s.t. $\forall \{u,v\} \in E$, $c(u) \neq c(v)$

Chromatic Number $\chi(G)$ = min # colors needed for proper coloring

★ Graph containing a triangle has min 3 colors

- Showing a k-coloring shows $\chi(G) \leq k$
- Show then that no $(k-1)$-coloring works

$\Rightarrow \chi(G) = $ exactly k.

C_n is cycle on n vertices:

$\chi(C_{2k}) = 2$, $\chi(C_{2k-1}) = 3$

K_n is clique on n vertices (Max # edges):

$\chi(K_n) = n$

4-Color Thm: Every Planar Graph is 4-colorable

Thm. If every vertex in G has deg. at most d, then $\chi(G) \leq d+1$

Lecture 11

For simple graphs, u & v connected if there is a path between them

As a binary relation, connectivity is:

- Reflexive: u & u always connected
- Transitive: u connect v & v connect w $\Rightarrow$ u connect w
- Symmetric: If u connected to v, then v connected to u.

In simple graphs, connectivity is an equivalence relation.

A graph is connected if it has 1 connected component

↳ Every vertex is connected to every other vertex in a connected component

G is Tree:

- Connected and acyclic
- Connected & every edge is cut edge
- Acyclic but adding edge creates cycle
- Every pair of vertices connected by unique path
- Connected, exactly $n-1$ edges
- Acyclic, exactly $n-1$ edges.

A cut edge is an edge s.t. removing it disconnects two vertices.

Bipartite Graph: Graphs w/ two sides

$G = (V,E)$ is bipartite if there are sets L & R s.t.

1). $L \cup R = V$

2). $L \cap R = \emptyset$

3). $E \subseteq \{\{l, r\} \mid l \in L, r \in R\}$

All edges have one endpoint in L and one endpoint in R.

G is bipartite iff G is 2 colorable.

G is bipartite iff every cycle has even length.

Trees are bipartite graphs

Bipartite Matching:
Set of $|L|$ edges s.t. every $\ell \in L$ appears once and every $r \in R$ appears ≤ 1 time.
↳ Can have more edges in R than L.
$|L| = |R|$ is perfect matching

Hall's Theorem:
There is matching iFF there are no bottlenecks
$(\forall S \subseteq L, |N(S)| \geq |S|)$
where $N(S) = \{r \in R \mid \exists \ell \in S \text{ s.t } \{\ell, r\} \in E\}$

Lecture 12: Evaluating Sums

For a summation, a closed-form solution is an algebraic equation with the same value as the summation

Pertubation Method:
- Combine a sum and a pertubation of that sum to get something useful.

ex. $S = \sum_{i=0}^{n-1} x^i$

$$S = 1 + x + x^2 + \dots + x^{n-1}$$
$$xS = x + x^2 + \dots + x^{n-1} + x^n$$
$$\Rightarrow S - xS = 1 - x^n \qquad S = \frac{1-x^n}{1-x}$$

Smart Grouping:
Identify groupings to make your life easier!

ex. $S = \sum_{i=1}^{n} i$

$S = 1 + 2 + \dots + n$ AND $S = n + n-1 + \dots + 1$

so $2S = n(n+1)$ and $S = \frac{n(n+1)}{2}$

Derivative Method: Use derivative of known closed forms to find new sums

ex. $S = m\sum_{i=0}^{n} i \cdot x^i$ Notice S looks like $\sum_{i=0}^{n} i x^{i-1} = \frac{d}{dx}\left(\sum_{i=0}^{n} x^i\right)$, where we know $\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$

$\rightarrow \sum_{i=0}^{n} i x^{i-1} = \frac{1-(n+1)x^n + nx^{n+1}}{(1-x)^2}$ $\therefore$ multiply both sides by x to get what we want!

Approximate by Integration Bounds (when no closed form)

if f is positive & increasing: $f(1) + \int_1^n f(x)\,dx \leq \sum_{i=1}^{n} f(i) \leq f(n) + \int_1^n f(x)\,dx$ flipped if decreasing

Lecture 13: Asymptotics

Notation	Test / Rule		
$f \sim g$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$		
$f \in O(g)$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} \in \mathbb{R}$	$\exists c \exists M. \forall x > M: \lvert f(x)\rvert \leq c\,g(x)$	$\leq$
$f \in o(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} = 0$		$<$
$f \in \Omega(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} \in (0, \infty]$	when $g \in O(f)$	$\geq$
$f \in \omega(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} = \infty$	when $g \in o(f)$	$>$
$f \in \Theta(g)$	$\lim_{n\to\infty} \frac{f(x)}{g(x)} \in (0, \infty)$	$f \in O(g)$ and $f \in \Omega(g)$	$=$

$f \in \omega(g) \Rightarrow f \notin O(g)$

$f \in o(g) \Rightarrow f \notin \Omega(g)$

Stirling's Formula:
$$n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$$

Lecture 14: Recurrences

Counting # of moves an algorithm takes:
- Let $T(n)$ be # moves for n elements.
- Find # moves for every step in the alg.
  (May be written as $T(\text{something})$).
- Combine to find $T(n)$, the open form solution.

Find closed form of $T(n)$ by:

Guess and Verify

Plug in small values of n

Guess pattern

Prove guess by induction

Plug & Chug (Expansion, Iteration, Brute Force)

- Plug in expansions of recursive terms where possible. Simplify
- Repeat until you can guess a pattern
- Prove guess by induction

Guess may still have recursive term. Find params s.t. that T(something) terms disappears. See lec ex.

Divide and Conquer Recurrence:

A D&C recurrence given by

$$T(x) = g(x) + \sum_{i=1}^{K} a_i T(b_i x) \text{ for } x > x_0 \text{ that satisfies:}$$

1). For all i, $a_i > 0$ and $0 < b_i < 1$

2). g is differentiable and polynomially bounded

3). X_0 is "big enough"

Thm Akra-Bazzi

If $T(n)$ has D&C recurrence, and

P s.t. $\sum_{i=1}^{K} a_i b_i^P = 1$ Increasing P decreases sum bc $b < 1$ ✱

Then, $T(n) \in \Theta\left(n^P + n^P \int_1^n \frac{g(x)}{x^{P+1}} dx\right)$

when P is not easy to find: find what P must be less than s.t. $\sum_{i=1}^{K} a_i b_i^P < 1$

Then, execute Akra-Bazzi and find $T(n)$ with P still in exponents

Then, find dominating term given the bound on P we found.

Thm. If $g(x) \in \Theta(x^t)$ and $t \geq 0$ and $\sum_{i=1}^{K} a_i b_i^t < 1$

Then, $T(x) \in \Theta(g(x))$

Master Theorem (Simpler cases of Akra-Bazzi for $K=1$) Here, $p = -\log_{b_1}(a_1)$

case 1: If $g(x) \in O(x^c)$ $c < p$

then $T(x) \in \Theta(x^P)$

case 2: If $g(x) \in \Theta(x^P)$

then $T(x) \in \Theta(x^P \log x)$

case 3: If $g(x) \in \Omega(x^c)$ $c > p$

then $T(x) \in \Theta(g(x))$

<u>Note on Boolean Matrix Mult.</u> For $n \times n$ matrices M & N,

$$\underset{i,j=1,\dots,n}{(M * N)(i,j)} := \bigvee_{k=1}^{n} \left(M[i,k] \wedge N[i,j]\right)$$, which returns 1 if either term is 1.

* Every DAG has one or more Topological Sorts
* Set of all minimal nodes forms an antichain.

<u>Def:</u> A cycle is a sequence $v_1, \dots, v_k, v_1$ s.t. $k \geq 3$, $v_1, \dots, v_k$ is a path, and $\{v_k, v_1\}$ is an edge.
→ $K_3 = C_3 =$ smallest cycle in undirected graph.