Lecture 8: Relations and Digraphs

A binary relation R from A to B consists of:

- Set A (Domain)
- Set B (Co Domain)
- Subset $graph(R) \subseteq A \times B$

$(x,y) \in graph(R)$ iFF x is "related" to y, xRy or $R(x,y)$

$graph(R)$ is set of all edges defined by the relation.

Can visualize w/ relation diagram.

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

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

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

Relation Composition: Combining / Chaining Relations

- 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.

Functions: Special Case of Binary Relation. aRb & $aRb' \Rightarrow b = b'$

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 $a \in A$.

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

A function is injective (1-to-1) if: each $b \in B$ has ≤ 1 incoming arrow (F^{-1} exists as func.)

A function is surjective if: each $b \in B$ has $\geq$ incoming arrow

A function is 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 $B \Rightarrow |A| = |B|$

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/ $E = graph(G)$
- Can have self-loops, isolated nodes, 2 edges in opp. directions, no edges.

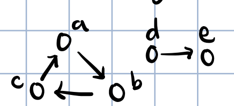
Walks: A sequence $v_1, \ldots, v_k \in V$ s.t. $\forall i = 1, \ldots, k$, $(v_i, v_{i+1}) \in E$

↳ k vertices are traversed in $k-1$ edges

Path: A walk in which all $v_1, \ldots, v_k$ are distinct

Adjacency Matrices

Consider graph G



$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

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

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

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

→ $u G^n v$ iff $\exists$ length n walk from u to v.

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

Note on Boolean Matrix Mult: For $n \times n$ matrices M & N,

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

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

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

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

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

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

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

Lecture 9: Partial Orders, Equivalence Relations

Directed Acyclic Graphs (DAGs):

- A directed graph with NO cycles.
  - ex. precedence relations (getting dressed)
- Minimal Elements: Elements with no "in-arrows" (Nothing comes before)
  - * After processing a minimal elt, you might uncover new minimal elts.
    - Can process all minimal elts at once.
- 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.
  - * Every DAG has one or more Topological Sorts
    - ↳ Find by "Greedy Algorithm": Repeatedly pick minimal elts, remove from graph & add to list.
- 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

Def'n: 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!

* Set of all minimal nodes forms an antichain.

1). # rounds required $\geq$ length of any chain

* Longest chain is called critical path ⇒ # rounds = length of critical path.

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

↳ if t = length of largest chain, can partition jobs into $\leq t$ antichains

Def'n A partition of set A is a set of nonempty subsets of A s.t. each elt in A is in exactly one block.

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

Orders:

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

↳ Has just one topological sort

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

Strict Partial Order is transitive & irreflexive. $<$ $\subset$
↳ Thm. R is S.P.O. iFF transitive and asymmetric.

Weak Partial Order is transitive, reflexive, and antisymmetric. $\leq$ $\subseteq$

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

- Equivalence class of x are all other elements y s.t. xRy
  ↳ Equivalence classes of R on A are blocks of partition of A.

Lecture 10: Simple Graphs: Degrees, Isomorphism, Coloring.

Simple Undirected Graphs:
- Undirected Edges
- At most one edge between vertexes (0 edges ok)
- Good for Symmetric Relationships

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

  * $\{u,v\} = \{v,u\}$

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

Handshaking Lemma

For every graph $G = (V, E)$, the sum of all degrees is twice the number of edges. $\sum_{v \in V} \deg(v) = 2|E|$

Thm. 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" ← all edges & non-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

Idea: Given a graph, assign a color to each vertex s.t. adjacent vertices have dift. colors using min. # colors

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 Theorem: Every planar graph is 4-colorable

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

Lecture 11 Simple Graphs: Connectivity, Trees, Bipartite Graphs

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. Directed Graphs don't have!

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

Trees: G is tree if:

- *G is connected and acyclic (does not have any cycles).*
- *G is connected, but removing any edge in G would disconnect the graph.*
- *G is acyclic, but adding any edge would create a cycle.*
- *Every pair of vertices of G is connected by a unique path.*
- *G is connected and has exactly $n-1$ edges.*
- *G is acyclic and has exactly $n-1$ edges.*

Another useful fact:

Theorem 2 (Lemma 12.11.3). *Every tree with at least 2 vertices has at least two leaves (i.e., at least two vertices with degree exactly 1).*

This is especially useful when doing Induction: start with a tree with $n+1$ vertices, remove a leaf (we know it has one!), apply the inductive hypothesis, and then reattach the leaf.

Def: 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.

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

↳ Every edge is cut edge in tree

Lemma: An edge e not cut edge iff e is on a cycle

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.

Thm. G is bipartite iff G is 2 colorable.

Thm. G is bipartite iff every cycle has even length.

Thm. Trees are bipartite graphs (bc color each layer alternating colors).

Bipartite Matching:

Goal: Match every $l \in L$ with unique neighbor $r \in R$ edge $\Leftrightarrow$ two nodes CAN be matched

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

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

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 $\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$

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

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

Approximating Sums with No Closed Form:

Method of Integration Bounds:

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

  * flipped if decreasing *

  $f(x)$ is what is inside the summation btw.

Lecture 13: Asymptotics

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

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

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

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

Lecture 14: Recurrences

Recursion is like induction on algorithms!

```
A(x):   if x small ... some stuff (base case)
        Else: ... some stuff
                  call A(x')  where x' is "smaller" than x. (recursive step)
              ... stuff
```

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(something)).
- Combine to find $T(n)$, the open form solution.

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

1). Guess and Verify
- Plug in small values of n
- Guess pattern
- Prove guess by induction

2). 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 $T(x) = g(x) + \sum_{i=1}^{k} a_i T(b_i x)$ for $x > x_0$ 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 $T(x) \in \Theta(g(x))$

RANDOM!

- Directed graphs have self-loops, which are cycles.
- Remember prove both directions for iFF

Some Common Sums:

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

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

$\sum k^2 = \frac{n(n+1)(2n+1)}{6}$

$\sum k^3 = \frac{n^2(n+1)^2}{4}$

Harmonic Series $H_n = \sum_{i=1}^{n} 1/i$

$\sum_{i=1}^{\infty} i x^{i-1} = \frac{1}{(1-x)^2}$

$\sum_{i=0}^{n} ar^i = \frac{ar^{n+1} - a}{r-1}$

Apply log to a product turns it into a sum:

$$\ln(n!) = \ln(1) + \ln(2) + \ldots + \ln(n) = \sum_{i=1}^{n} \ln(i).$$

Proofs By Contradiction when you need to prove something can't happen.

Quiz 2 Review 6.042

Lecture 8 – Relations and Digraphs (Directed Graphs)

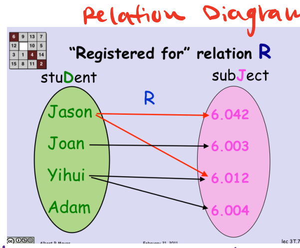
Relations + Functions

- Binary Relations – associates elements of domain (Left) with elements of codomain (Right)
  - Notation: J is "registered for" (R) 6.042
    - Jason R 6.042 — infix
    - R(Jason, 6.042) — prefix
    - (Jason, 6.042) ∈ R
    - (Jason, 6.042) ∈ graph(R)

"less than" relates pairs of #'s
"subset" relates pairs of sets.

subset graph(R) $\subseteq A \times B$
↳ pairs of elements $a \in A, b \in B$ (a,b).

Relation Diagram



domain codomain

- $R(X)$ ::= everything R relates to things in X, endpoints of arrows from points in X
  $\{j \in J \mid \underbrace{\exists d \in X.\ d R j}_{\text{arrow from } X \to j}\}$

ex. R(Jason), R({Jason, Yihui})

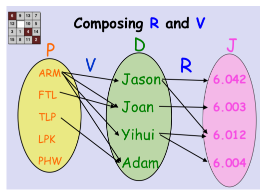
- R^{-1} (inverse image): dRj IFF $jR^{-1}d$,
  - $D \subseteq R^{-1}(J) \Rightarrow$ every student is registered

Domain & Codomain swapped and $\text{graph}(R^{-1}) := \{(b,a) \in B \times A \mid a R b\}$

ex. $R^{-1}(6.012)$

- Composition: $R \circ V(X) ::= R(V(X))$ * V happens first, then R
  Combining Relations
  - $R \circ V$ ::= "prof has advisee registered for"
  - $p\,(R \circ V)\,j$ ::= prof p has an advisee in subject j
  - Ex. ARM $(R \circ V)$ 6.012 b/c ARM V Yinui and Yinui R 6.012

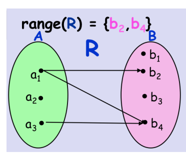
$\text{graph}(T \circ R) := \{(a,c) \mid \exists b.\ \underset{\text{1st arrow}}{(aRb)} \wedge \underset{\text{2nd arrow}}{(bTc)}\}$



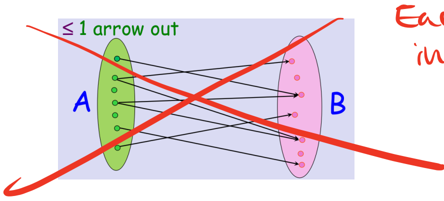
$R \circ V$
A relation from P to J

- Set operations on relations
  - $T \cap (R \circ V) = \varnothing$, Professors should not teach advisees
    $R \circ V \subseteq \overline{T}$

- Range(R) ::= elements with arrows coming in
  all elements in Codomain connected to Domain.



- Function: relation which associates each element, a, of A with at most 1 element of B (1 arrow out)
  (aFb AND aFb') IMPLIES b=b'



Each element in A points to at most on element in B
↳ one x value in a func. can point to at most one y value.

Relational Mapping Properties

OUT ARROWS

- function: ≤ 1 arrow out
- total relation: ≥ 1 arrow out   ex. $\frac{1}{x-y}$ is not total, $g(r,r)$ DNE
  - → R is total IFF $A = R^{-1}(B)$
- total and function: exactly 1 arrow out

IN ARROWS

- surjection: ≥ 1 arrow in   (see if $\mathbb{R}$ is possible for outputs)
  - → R is a surjection IFF $R(A) = B$
- injection: ≤ 1 arrow in   (check if multiple inputs → 1 output)
- bijection: exactly 1 arrow out, exactly 1 arrow in
  - → implies $|A| = |B|$
  - → must be: surjection, injection, total, function

If f is total, then f is bijection IFF f is injective and surjective.

Directed Graphs (Digraphs)

- Digraphs:
  - a set, V, of vertices
  - a set, $E \subseteq V \times V$ of directed edges
    - $(v, w) \in E$
    - notation: $v \to w$
  - a digraph with vertices V is the same as a binary relation on V

  -can have self loops, isolated nodes, 2 edges in opposite dir, no edges
- Walks: follow successive edges k vertices are traversed in $k-1$ edges
- Paths: walk thru vertices without repeating vertex... a more specific walk
- Adjacency Matrix

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

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

Connected vertices

- Lemma: the shortest walk between two vertices is a path!
- length n walk relation
  - $v\, G^n\, m$ IFF $\exists$ length n walk from v to m,
    G^n is the length n walk relation for G
  - Lemma: $G^m \circ G^n = G^{m+n}$
- Matrices & Composition : $A_G ::=$ adjacency matrix for G
  - Lemma: $A_{G \circ H} = A_H \odot A_G$
    ??   boolean matrix prod (using or)
- Walk Relation G^* : G^* is walk relation of G $u\, G^*\, v$ IFF $\exists$ walk u to v (u is connected to v)

G^*, G^+ ?

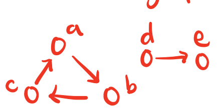
Lemma: $\forall n \geq 0$, $A_{G^n \circ G} = A_{G^n} * A_G = A_{G^{n+1}}$

Boolean Matrix Mult: $(M * N)(i,j) := \bigvee_{k=1}^{n} (M[i,k] \wedge N[k,j])$, $i,j = 1, \dots, n$

returns 1 if @ least one of our tries $N[i,k] \wedge N[i,j]$ is 1.

$\forall n \geq 1$, $A_{G^n} = (A_G)^n$

Consider graph G



$$\begin{array}{c|c} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

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

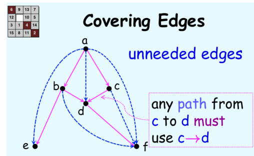
<u>Connectivity</u>: u is connected to v if $\exists$ walk from u to v.

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

Lecture 9 – Partial Orders, Equivalence Relations

Directed Acyclic Graphs (DAGs)

- indirect prerequesite: there is a sequence of prereq's from u to v
  - → aka there is a positive length walk from u to v ($u\ D^+\ v$)
- closed walk: start + end on same vertex
- cycle: closed walk whose only repeat vertex is its start
  - → 1 vertex = length 0 cycle
- Lemma: shortest positive length closed walk from v is positive length cycle from v
  - → proof: like walk / path
- directed acyclic graph (DAG): has no positive length cycle (or closed walk)
  - → ex. prerequisite graph, $\subset$ $\{a, b\} \rightarrow \{a, b, d\}$
  - → covering edge: needed edge



DAG's & Scheduling

- a minimal subject: no prerequisites
  - → nothing → d, $\forall x \in A$ s.t. $\forall y \neq x$, $y \npreceq x$
- minimum subject: earliest of all (an indirect prereq)
- "greedy": take max subjects each term
- antichain: set of subjects with no indirect prereqs, "incomparable" ⇗ if neither $a \preceq b$ nor $b \preceq a$
  - → u is incomparable to v iff no path from u to v and no path from v to u
- topological sort: take classes by layer, every finite poset has a topological sort
- chain: sequence of subjects that must be taken in order
  - → max chain length ≤ # terms to graduate

Partial Orders

- strict partial orders: transitive + asymmetric
  - → ex: • $\subset$ on sets
    - • "indirect prerequisite"
    - • less than, <, on real #s
  - → R is a SPO IFF $R = D^+$ for some DAG
- linear ^path-total orders: given any 2 elements, 1 will be "bigger than" the other
  - → ex: < or ≤ on reals
  - → no incomparable elements — x R y or y R x
  - → whole partial order is a chain
- topological sort: turns a partial order into a linear order ... in a way that is consistent w/ partial order
- weak partial order: same as strict partial order R, except that a R a always holds
  - → poset
  - → ex. ≤, ⊆
  - → transitive, antisymmetric, reflexive (all elements)
  - → Thm: R is a WPO iff $R = D^*$ for some DAG D
- Notation: $a \preceq b$ – a comes before b in partial order
  - → Poset: partially ordered set $(A, \preceq)$
  - → same as a DAG (except self-loops)

Summary of Relational Properties (10.11 in txtbk) *see Rec 9 notes

* Relation $R: A \to A$ is the same as a digraph with vertices A

"R is ___ when"

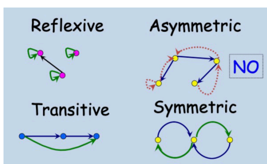
- Reflexivity: $\forall x \in A.\ x R x$ (every vertex has self loop)
- Irreflexivity: $\text{NOT}[\exists x \in A.\ x R x]$ (no self-loops in R)
- Symmetry: $\forall x, y \in A.\ x R y$ IMPLIES $y R x$ (if edge $x \to y$, then $\exists\ y \to x$ in R)
- Asymmetry: $\forall x, y \in A.\ x R y$ IMPLIES $\text{NOT}(y R x)$ (at most 1 directed edge btwn any 2 vertices, no self-loops)
- Antisymmetry: $\forall x \neq y \in A.\ x R y$ IMPLIES $\text{NOT}(y R x)$ (" " ", may be self-loops)
  or $\forall x, y \in A.\ (x R y$ AND $y R x)$ IMPLIES $x = y$
- Transitivity: $\forall x, y, z \in A.\ (x R y$ AND $y R z)$ IMPLIES $x R z$ (if path from $u \to v$, edge from $u \to v$)
- Linear: $\forall x \neq y \in A.\ (x R y$ or $y R x)$ (given any 2 vertices, $\exists$ some edge)

Partial Orders:

- Strict Partial Order: R is a strict partial order IFF
  R is transitive and irreflexive IFF
  R is transitive and asymmetric IFF
  it is the positive length walk relation of a DAG
  ex. • $\subset$ on sets • "indirect prerequisite" • $<$ on real #s
- Weak Partial Order: R is a weak partial order IFF
  R is transitive and anti-symmetric and reflexive IFF
  R is the walk relation of a DAG
  ex. $\leq, \subseteq$
- Equivalence Relation: R is an equivalence relation IFF
  R is reflexive, symmetric, transitive IFF
  R equals the in-the-same-block-relation for some partition of domain(R)
  ex. $=, \equiv$, same size, same color

Equivalence Relations

- two-way walks: $u G^* v$ AND $v G^* u$, u and v are strongly connected
- equivalence relations: transitive, symmetric, reflexive
  - → R is equiv. rel. IFF R is strongly connected relation of diagraph
  - → ex. $=, \equiv$, same size/color
- Graphical Properties of Relations



- Representing equivalences
  - → for total function $f: A \to B$, define relation $\equiv_f$ on A: $a \equiv_f a'$ iff $f(a) = f(a')$
  - → Thm: R on set A is equiv. relation IFF R is $\equiv_f$ for some $f: A \to B$
  - → $\equiv (\text{mod } n)$ is $\equiv_f$ (where $f(k) ::= \text{rem}(k, n)$)
  - → for partition Π of A, def. relation $\equiv_\Pi$ on A: $a \equiv_\Pi a'$ IFF a, a' are in same block of Π
  - → Thm: R on A is an equiv. rel. IFF R is $\equiv_\Pi$ for some partition Π of A

Posets (Partially Ordered Sets) - relation that's transitive, antisymmetric, reflexive

- Hasse diagram – a way of representing a poset $(A, \preceq)$ as a DAG
- chain – a subset $C \subseteq S$ in poset $(S, \preceq)$. $\forall x, y \in C$, either $x \preceq y$ or $y \preceq x$
- antichain – a subset $A \subseteq S$. $\forall x, y \in A$, $x \neq y$, neither $x \preceq y$ nor $y \preceq x$
- topological sort – total order of all the elements such that if $a_i \preceq a_j$ in the partial order, then a_i precedes a_j in the total order

Lecture 10: simple Graphs - Isomorphism, Coloring

Simple Graphs: Degrees

- simple graph G - consists of
  - nonempty set V of vertices
  - a set E of edges such that each edge has 2 endpoints in V
  - undirected edges
- vertex degree - # of incident edges (that touch it)
- Handshaking Lemma - $2|E| = \sum_{v \in V} deg(v)$ (sum of degrees = twice # edges)

Simple Graphs: Isomorphism

- isomorphic: $\exists$ bijection $f: V_1 \to V_2$ with $u-v$ in E_1 IFF $f(u)-f(v)$ in E_2
  - aka there is an edge-preserving matching of their vertices (that is bijective)
  - NOT isomorphic if any properties differ:
    - # nodes
    - # edges
    - degree distributions
    - length of any cycles that exist

Simple Graphs: Coloring

- planar graph: if it is possible to find a planar drawing of that graph (a drawing where the edge-lines don't intersect each other)
- coloring: assignment of colors to each vertex in a graph G so no edge has the same coloring on both ends
  - Any planar map is 4-colorable.
  - Note: 7-colorable ≠ not 6-colorable
- Chromatic number $\chi(G)$: minimum # of colors to validly color G
  - G is k-colorable iff $\chi(G) \le k$
  - for a complete graph K_n (edge btwn every 2 vertices), $\chi(K_n) = n$
  - $\chi(C_{even}) = 2$
  - $\chi(C_{odd}) = 3$
  - all degrees $\le k$ implies $\chi(G) \le k+1$

Graph Proof: Inductive Step

① Start with G(n+1) that satisfies desired constraints
② Remove some vertex (or edge) to get G(n)
③ Prove G(n) satisfies desired constraints
④ Apply IH to G(n)
⑤ Add vertex back and show P(n+1) holds

Solution. *Proof.* By induction on the number of vertices n. The induction hypothesis is

$$P(n) := \text{all } n\text{-vertex graphs } G \text{ with width one are forests.}$$

Base case: $(n = 1)$. A graph with one vertex is acyclic and therefore is a forest.

Induction step. Assume that $P(n)$ is true for some $n \ge 1$ and let G be an $(n+1)$-vertex graph with width one. We need only show that G is acyclic.

The vertices of G can be listed with each vertex adjacent to at most one vertex earlier in the list. Let v be the last vertex in the list. Since *all* the vertices adjacent to v appear earlier in the list, it follows that $deg(v) \le 1$.

Now removing a vertex won't increase width, so $G \setminus v$ (G without v) still has width one, so it is acyclic by Induction Hypothesis. But no degree-one vertex is in a cycle, so adding v back to $G \setminus v$ will not create a cycle. Hence G is acyclic, as claimed.

This proves $P(n+1)$ and completes the induction step.

(Rec 11)

Lecture 11: Bipartite Matching, Trees

Simple Graphs: Connectivity

- A graph is connected iff all its vertices are connected to each other
  - has only 1 connected component
- 2 vertices are connected iff there is a path btwn them
- Connected component: $\underbrace{E^*}_{\text{walk relation}}(v)$

Trees

- Tree definitions: (more below)
  - A connected graph with no cycles
  - Every edge is a cut edge
  - A connected graph that is edge minimal
  - connected graph with n vertices and n-1 edges
  - An edge-maximal acyclic graph
  - Graph with a unique path btwn any 2 vertices
- Cut edge - removing the edge disconnects 2 vertices
  - Lemma: an edge is not a cut edge iff it is on a cycle
- Leaf: vertex of degree at most 1
- Trees:

Theorem 1 (Theorems 12.11.4 and 12.11.6). *For an undirected graph G with $n = |V(G)|$ vertices, the following statements are all equivalent to G being a tree:*

- *G is connected and acyclic (does not have any cycles).*
- *G is connected, but removing any edge in G would disconnect the graph.*
- *G is acyclic, but adding any edge would create a cycle.*
- *Every pair of vertices of G is connected by a unique path.*
- *G is connected and has exactly $n - 1$ edges.*
- *G is acyclic and has exactly $n - 1$ edges.*

Another useful fact:

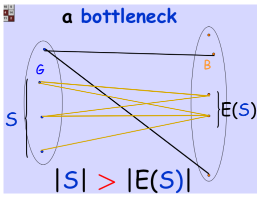
Theorem 2 (Lemma 12.11.3). *Every tree with at least 2 vertices has at least two leaves (i.e., at least two vertices with degree exactly 1).*

This is especially useful when doing Induction: start with a tree with $n + 1$ vertices, remove a leaf (we know it has one!), apply the inductive hypothesis, and then reattach the leaf.

- Tree coloring:
  - $\chi(G) = 2$ (all cycles, if any, are even len)

Bipartite Matching

- Bipartite graph: vertices can be partitioned into 2 blocks, $L(G)$ and $R(G)$ such that every edge has 1 endpoint in $L(G)$ and the other in $R(G)$
  - graph is bipartite iff it is 2-colorable
- bottleneck: a set S of girls without enough boys, $|S| > |E(S)|$



Hall's Theorem

- bipartite match: a total injective function $L(H) \to R(H)$
  - matching: any pairing of some subset of L with R such that no element is paired w/ > 1 element
  - perfect: includes all elements
- Hall's Theorem: If there are no bottlenecks, then there is a match, aka $\boxed{\text{If } |S| \leq |E(S)| \text{ for all sets } S \subseteq L(H)}$, then there is a match.
  Hall's condition
- Thm 12.5.6. - If G is a degree-constrained bipartite graph, then there is a matching that covers $L(G)$.
  - degree-constrained: if min degree (L) $\geq$ max degree (R) [Rec 11]
- Proof of Hall's Thm

  Case 1) $|S| = |E(S)|$
  - Lemma. If S a set of girls with $|S| = |E(S)|$, then no bottlenecks btwn $\bar{S}$ and $\overline{E(S)}$ either
  - so, no bottlenecks in $(S, E(S))$ and none in $(\bar{S}, \overline{E(S)})$

  Case 2) $|S| < |E(S)|$
  - match g with b
  - removing b still leaves $|S| \leq |E(S)|$, so no bottlenecks.
  - by induction, can match remaing g + b

2 Hall's Matching Theorem

We use the notation $G = (L, R, E)$ for a bipartite graph with vertices $V = L \cup R$, where L and R are disjoint sets representing the "left" vertices and "right" vertices, respectively, and where every edge has exactly one endpoint in L and one endpoint in R.

A ***matching*** in G is a subset of edges $M \subseteq E$ so that every vertex in G lies on *at most* one edge from M.

If M is a matching and every vertex $v \in L$ belongs to an edge in M, we say that ***M covers L***. (Define ***covers R*** similarly.) If M covers both L and R, we call M a ***perfect matching*** (note that this requires $|L| = |R|$!).

A subset $S \subseteq L$ is called a ***bottleneck*** in G if it connects to strictly fewer than $|S|$ neighbors in R, via edges in E.

We have the **quick observation** that having a bottleneck immediately shows that there can't be a matching that covers L: if $S \subseteq L$ is a bottleneck, then the vertices in S collectively have fewer than $|S|$ options to choose from, so someone from S will always be left out. So avoiding bottlenecks is a **necessary condition** for having a matching that covers L.

The **more surprising** fact is that this condition is **also sufficient**:

Theorem 3 (Hall's Matching Theorem (Theorem 12.5.4)). *A bipartite graph $G = (L, R, E)$ has a matching that covers L* ***if and only if*** *no subset $S \subseteq L$ is a bottleneck.*

Lecture 12: Sums

Example: Pricing an annuity

def: A n-year \$m - payment annuity pays \$m at start of each year for n years

Assumption : fixed interest rate P

- \$1 today = \$(1+p) in 1 year
- $\$\frac{1}{1+p}$ today = \$1 in 1yr
- $\$\frac{1}{(1+p)^2}$ today = \$1 in 2yrs

⋮

current value	payments
\$m	\$m now
$\$m/(1+p)$	\$m in 1yr
⋮	⋮
$\$m/(1+p)^{n-1}$	\$m in n-1 yrs

$$V = \sum_{i=0}^{n-1} m/(1+p)^{n-1} = m \sum_{i=0}^{n-1} x^i ; x = \frac{1}{1+p}$$

$$\boxed{\sum_{i=0}^{n-1} x^i = \frac{1-x^n}{1-x}}$$

Perturbation: line up terms, get cancellation

$$\begin{aligned} S &= 1+x+x^2 \dots \quad x^{n-1} \\ -xS &= \quad x+x^2 \dots \quad x^{n-1}+x^n \\ \hline (1-x)S &= 1-x^n \Rightarrow S = \frac{1-x^n}{1-x} \end{aligned}$$

Claim: If $n=\infty$, $V = m\left(\frac{1+p}{p}\right)$

Proof: $\lim_{n\to\infty}\left(\frac{1}{1+p}\right)^n = 0 \Rightarrow \lim_{n\to\infty} V = \frac{m}{1-\frac{1}{1+p}} = \left(\frac{1+p}{p}\right)m$ □

Corollary: If $|x|<1$, $\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$ e.g. $1+\frac{1}{2}+\frac{1}{4}\dots = \frac{1}{1-\frac{1}{2}} = 2$

$1+\frac{1}{3}+\frac{1}{9}\dots = \frac{1}{1-\frac{1}{3}} = \frac{3}{2}$

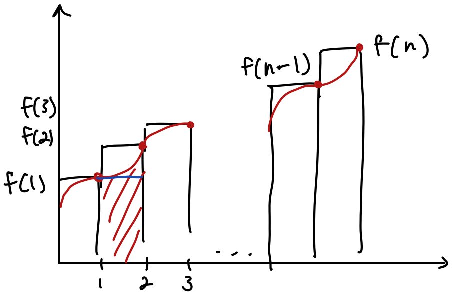
Derivative: $\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$ $\int(\Sigma x^i) = \Sigma \frac{x^{i+1}}{i+1}$

$\Downarrow \frac{d}{dx}$

$$x\left(\sum_{i=0}^{n-1} i x^{i-1} = \underbrace{\frac{-(n+1)x^n(1-x)+(1-x^{n+1})}{(1-x)^2}}_{f(x)}\right)$$

$$\sum_{i=1}^{n-1} i x^i = f(x)\cdot x$$

Integration Bounds: If f is positive and increasing



$$I = \int_1^n f(x)\,dx$$

If f is weakly increasing + $S ::= \sum_{i=1}^{n} f(i)$,

$$I + f(1) \leq S \leq I + f(n)$$

If f is weakly decreasing, then

$$I + f(n) \leq S \leq I + f(1)$$

Summation Rules

1. $\sum_{i=1}^{n} c = cn$
2. $\sum_{i=1}^{n} c \cdot a_i = c \sum_{i=1}^{n} \cdot a_i$
3. $\sum_{i=1}^{n} (a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i$
4. $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
5. $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$
6. geometric sum $\sum_{i=0}^{n-1} x^i = \frac{1-x^n}{1-x}$ $\quad \sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}$ $\quad \sum_{k=j}^{n} 2^k = 2^{n+1} - 2^j$

* review double sums

Lecture 13: Asymptotics

Stirling's Formula:

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n}\, e^{\varepsilon(n)} \quad \text{where} \quad \frac{1}{12n+1} \le \varepsilon(n) \le \frac{1}{12n}$$

$$100! \ge \left(\frac{100}{e}\right)^{100} \sqrt{200\pi}\, e^{\frac{1}{1201}}$$
$$100! \le \left(\frac{100}{e}\right)^{100} \sqrt{200\pi}\, e^{\frac{1}{1200}}$$

$$n! \sim \left(\frac{n}{e}\right) \sqrt{2\pi n}$$

Asymptotic Notation:

Name	Notation	Explanation	Summary
"tilde"	$f(x) \sim g(x)$	$\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$	
"O" / "Big O"	$f(x) = O(g(x))$	$\lim_{x\to\infty} \frac{f(x)}{g(x)} < \infty$	$\le$
"omega"	$f(x) = \Omega(g(x))$	$\lim_{x\to\infty} \left\lvert\frac{f(x)}{g(x)}\right\rvert > 0$	$\ge$
"theta"	$f(x) = \Theta(g(x))$	$\lim_{x\to\infty} \left\lvert\frac{f(x)}{g(x)}\right\rvert > 0, < \infty$	$=$
"little o"	$f(x) = o(g(x))$	$\lim_{x\to\infty} \frac{f(x)}{g(x)} < \infty$	$<$
"little ω"	$f(x) = \omega(g(x))$	$\lim_{x\to\infty} \left\lvert\frac{f(x)}{g(x)}\right\rvert > 0$	$>$

if Θ, then O and Ω

Perturbation Method (R 13)

$$S = \sum_{i=0}^{\infty} \frac{F_i}{2^i}$$

$$S = F_0 + \frac{F_1}{2} + \frac{F_2}{2^2} \dots + \frac{F_n}{2^n}$$
$$\frac{S}{2} = \frac{F_0}{2} + \frac{F_1}{4} \dots + \frac{F_n}{2^{n+1}}$$

$$S + \frac{S}{2} = F_0 + \sum_{i=0}^{\infty} \frac{F_i + F_{i+1}}{2^{i+1}} = F_0 + \sum_{i=0}^{\infty} \frac{F_{i+2}}{2^{i+1}} = F_0 + 2\left(S - \frac{F_0}{2^0} - \frac{F_1}{2^1}\right)$$

$$\sum \frac{F_{i+2}}{2^{i+1}} \Rightarrow \frac{F_2}{2} + \frac{F_3}{2^2}$$

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

Lecture 13:

Lecture 14: Divide + Conquer Recurrences

(Review Rec)

- Def: T_n = min # moves needed for n discs

Guess + Verify (aka substitution)

Guess: $T_n = 2^n - 1$

Verify: by induction

$P(n)$: $T_n = 2^n - 1$

Base Case: $n=1$, $T_1 = 1 = 2^1 - 1 = 1$ ✓

Ind Step: Assume $T_n = 2^n - 1$ to prove $T_{n+1} = 2^{n+1} - 1$

$T_{n+1} = 2T_n + 1 = 2(2^n - 1) + 1 = 2^{n+1} - 2 + 1 = 2^{n+1} - 1$ ✓

Merge Sort

To sort $n > 1$ items $x_1, x_2, \ldots, x_n$ (n = power of 2)

1. sort $x_1, x_2, \ldots x_{n/2}$ & $x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$ recursively
2. merge

Def. $T(n)$ = # comparisons used to sort n numbers

merging takes $n-1$ comparisons (worst case)

recursive sorting takes $2T(\frac{n}{2})$ comparisons

$\Rightarrow T(n) = n - 1 + 2T(n/2)$

Plug & Chug (aka Expansion, Iteration, Brute force)

$$
\begin{aligned}
T(n) &= n-1 + 2T(n/2) \\
&= n-1+2\left(\tfrac{n}{2}-1+2T\left(\tfrac{n}{4}\right)\right) = n-1+n-2+4T\left(\tfrac{n}{4}\right) \\
&= n-1+n-2+4\left(\tfrac{n}{4}-1+2T\left(\tfrac{n}{8}\right)\right) = n-1+n-2+n-4+8T\left(\tfrac{n}{8}\right) \\
&= n-1+n-2 \ldots + n-2^{i-1} + 2^i T\left(\tfrac{n}{2^i}\right) \qquad i = \log n \\
&= n-1+n-2 \ldots \; n-2^{\log n - 1} + 2^{\log n} \cancel{T\left(\tfrac{n}{2^{\log n}}\right)}^{\,0} \\
&= \sum_{i=0}^{\log n - 1} (n - 2^i) = n\log n - (2^{\log n} - 1) = n \log n - n + 1
\end{aligned}
$$

$$
\left\{
\begin{array}{lll}
T(n) = 2T(\underline{n-1}) + 1 & \Rightarrow & T(n) \sim 2^n \\
T(n) = 2T(\underline{\tfrac{n}{2}}) + n - 1 & \Rightarrow & T(n) \sim n\log n \\
S(n) = 2S(\tfrac{n}{2}) + 1 & \Rightarrow & S(n) \sim n
\end{array}
\right.
$$

} divide + conquer

Divide + Conquer Recurrences have the form

$$T(x) = a_1 T(b_1 x + \varepsilon_1(x)) + a_2 T(b_2 x + \varepsilon_2(x)) + \ldots + a_k T(b_k x + \varepsilon_k(x)) + g(x) \quad \text{for } x \geq x_0$$

$$T(x) = \Theta(1) \quad \text{for } x \leq x_0$$

where: $a_i > 0$, $0 < b_i < 1$, k fixed

$|\varepsilon_i(x)| \leq O\left(\frac{x}{\ln^2 x}\right)$

$|g(x)| \leq x^c$ for some c

Akra-Bazzi Theorem

Theorem 1 (Akra-Bazzi, strong form). *Suppose that:*

$$T(x) = \begin{cases} \text{is defined} & \text{for } 0 \le x \le x_0 \\ \displaystyle\sum_{i=1}^{k} a_i T(b_i x + h_i(x)) + g(x) & \text{for } x > x_0 \end{cases}$$

where:

- $a_1, \ldots, a_k$ *are positive constants*
- $b_1, \ldots, b_k$ *are constants strictly between 0 and 1*
- x_0 *is "large enough" in a technical sense we leave unspecified*
- $|g'(x)| = O(x^c)$ *for some* $c \in \mathbb{N}$
- $|h_i(x)| = O(x/\log^2 x)$

Then:

$$T(x) = \Theta\left(x^p\left(1 + \int_1^x \frac{g(u)}{u^{p+1}}\, du\right)\right)$$

where p *satisfies the equation* $\sum_{i=1}^{k} a_i b_i^p = 1$.

example:

The only difference between the strong and weak forms of Akra-Bazzi is the appearance of this $h_i(x)$ term in the recurrence, where $h_i(x)$ represents a small change in the size of the subproblems ($O(x/\log^2 x)$). Notice that, despite the change in the recurrence, the solution $T(x)$ remains the same in both the strong and weak forms, with no dependence on $h_i(x)$! In algorithmic terms, this means that *small* changes in the size of subproblems have no impact on the asymptotic running time. **Example:** Let's compare the Θ bounds for the following

divide-and-conquer recurrences.

$$T_a(n) = 3T\left(\frac{n}{3}\right) + n \qquad\qquad T_b(n) = 3T\left(\left\lfloor\frac{n}{3}\right\rfloor\right) + n$$

For $T_a(n)$ we have $a_1 = 3$, $b_1 = 1/3$, $g(n) = n$, $p = 1$. For $T_b(n)$ we have the same parameters as for $T_a(n)$, plus $h_1(n) = \lfloor n/3 \rfloor - n/3$. Using the strong Akra-Bazzi form, the $h_1(n)$ falls out of the equation:

$$T_a(n) = T_b(n) = \Theta(n(1 + \int_1^n \frac{u}{u^2} du)) = \Theta(n \log n).$$

The addition of the ceiling operator changes the value of $n/3$ by at most 1, which is easily $O(n/\log^2 n)$. So floor and ceiling operators have no impact on the asymptotic solution to a recurrence.