CS 381 – Spring 2021

Week 1, Lecture 2
Part 1

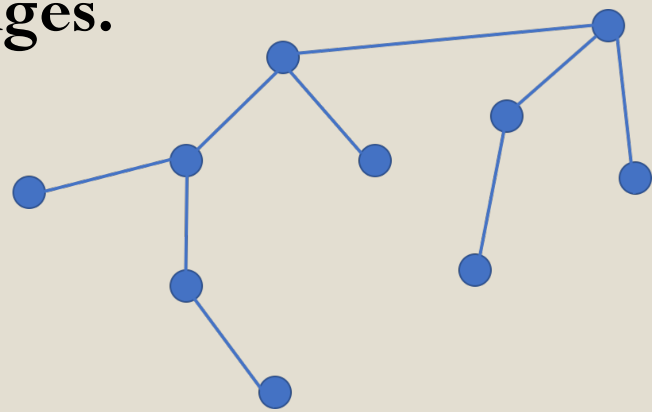
Reminder: What is active participation?

- ***Working with your study group***
  - For studying the material, working on assignments, preparing for exams, solving extra problems that you are interested in.
  - Questions should be brought to TAs and instructors only **after** discussing them with your study group.

- ***Give meaningful answers on Piazza.***
  - Also point students to questions already answered. Reading questions and answers can increase understanding.

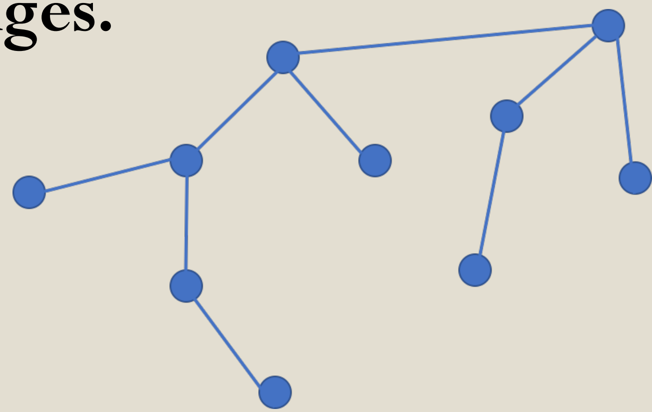
Claim: Any tree with n nodes has n-1 edges.

Note: Edges are undirected; i.e. no orientation.



Claim: Any tree with n nodes has n-1 edges.

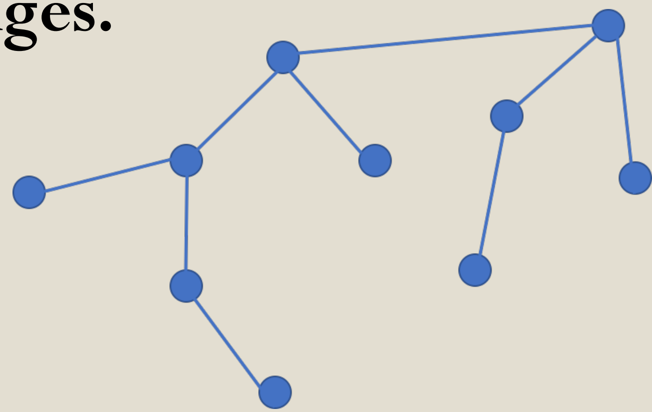
Note: Edges are undirected; i.e. no orientation.



When can each node of a tree have degree ≥ 2 ?

Claim: Any tree with n nodes has n-1 edges.

Note: Edges are undirected; i.e. no orientation.



When can each node of a tree have degree ≥ 2 ?

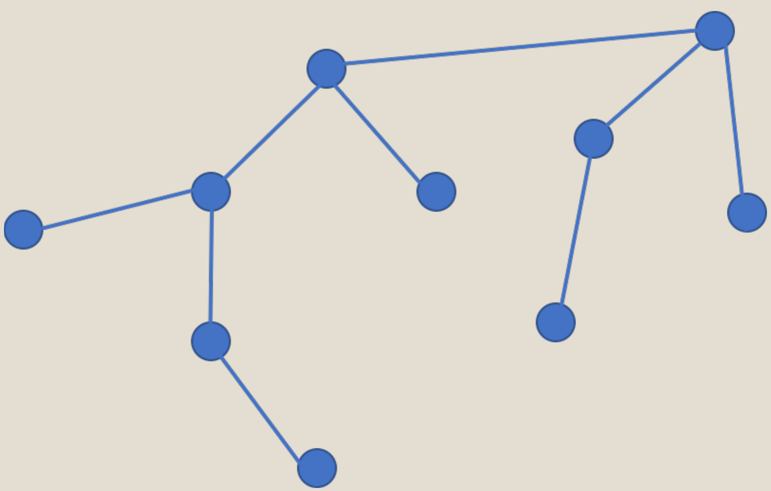
Recall: A tree must have a node with degree 1 when n>1.

Proof: exercise.

Claim: Any tree with n nodes has n-1 edges.

Proof: by induction.

- Base case:
- Inductive hypothesis:
- Inductive step:

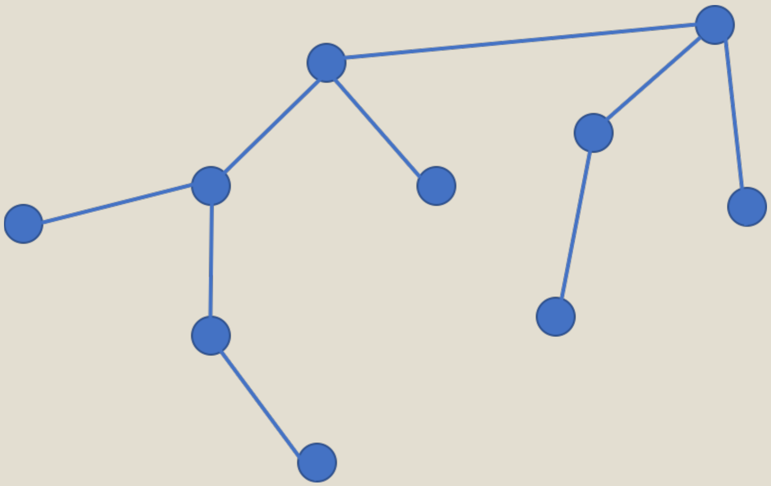


Recall: A tree must have a node with degree 1 when n>1.

Claim: Any tree with n nodes has n-1 edges.

Proof: by induction.

- Base case: $n = 1 \Rightarrow$ no edges.
- Inductive hypothesis:
- Inductive step:

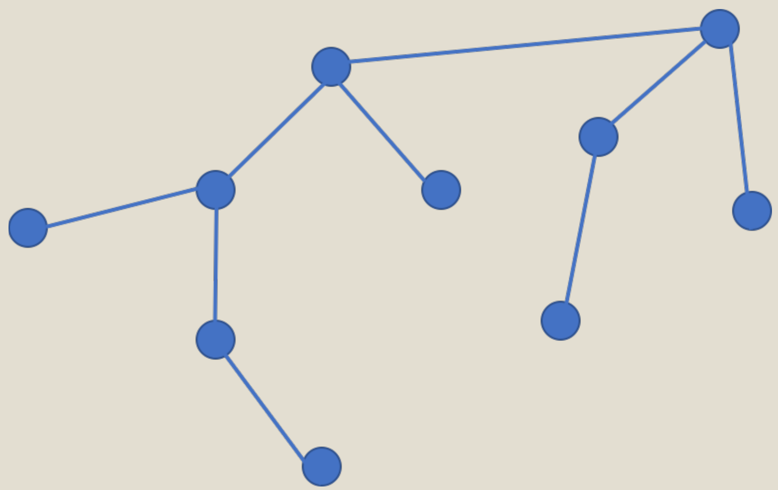


Recall: A tree must have a node with degree 1 when n>1.

Claim: Any tree with n nodes has n-1 edges.

Proof: by induction.

- Base case: $n = 1 \Rightarrow$ no edges.
- Inductive hypothesis: claim true for all trees of size $\leq n - 1$.
- Inductive step:

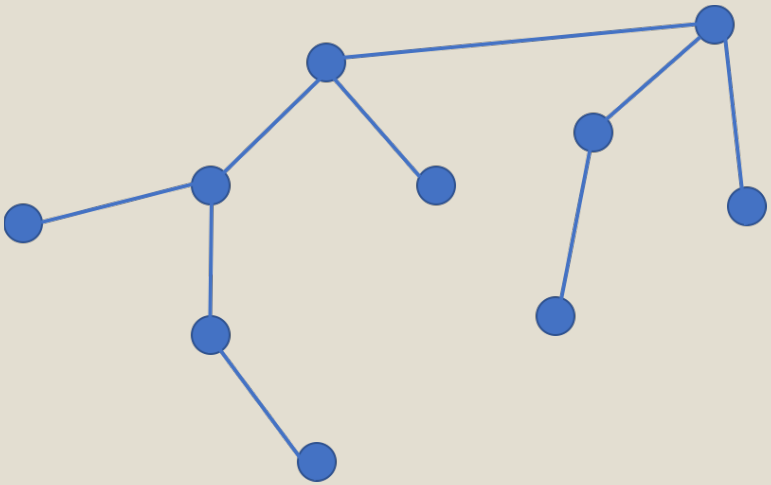


Recall: A tree must have a node with degree 1 when n>1.

Claim: Any tree with n nodes has n-1 edges.

Proof: by induction.

- Base case: $n = 1 \Rightarrow$ no edges.
- Inductive hypothesis: claim true for all trees of size $\leq n - 1$.
- Inductive step: take any tree T of size n. It can be decomposed into a tree T' of size n-1 and a separate node x, such that x is a leaf in T.



Recall: A tree must have a node with degree 1 when n>1.

Tribonacci sequence

- Basis: $f(1) = f(2) = f(3) = 1$.
- Recurrence: $f(n) \;=\; f(n-1) + 3f(n-2) + 9f(n-3)$ for $n > 3$.

Tribonacci sequence

- Basis: $f(1) = f(2) = f(3) = 1$.
- Recurrence: $f(n) \;=\; f(n-1) + 3f(n-2) + 9f(n-3)$ for $n > 3$.

Claim: $\mathbf{f}(\boldsymbol{n}) \leq \mathbf{2} \cdot \mathbf{3}^{\boldsymbol{n}-\mathbf{2}}$ for all n>1.

Tribonacci sequence

- Basis: $f(1) = f(2) = f(3) = 1$.
- Recurrence: $f(n) = f(n-1) + 3f(n-2) + 9f(n-3)$ for $n > 3$.

Claim: $\mathbf{f}(\boldsymbol{n}) \leq \mathbf{2} \cdot \mathbf{3}^{\boldsymbol{n}-\mathbf{2}}$ for all n>1.

Proof: by induction.

- Basis: what are the base cases?
- Inductive hypothesis: what are the IHs?
- Inductive step: how does the proof work inductively?

Analysis of Algorithms

Worst case analysis

- in an asymptotic sense, the maximum time the algorithm takes on any input of size n.

Average case analysis

- expected time; often meaningful
- may need assumptions on the statistical distribution of input data

Best case analysis

- does not mean much; generally easy to determine

For some algorithms, the three bounds are identical

- Means performance does not depend on the value of the data
- For some algorithms, average case performance is only known experimentally.

What do we count?

- Time and space
  - time in terms of number of basic operations on basic data types
- Ignore machine dependent factors, but remain realistic
- Random Access Model (RAM)
  - no concurrency
  - count instructions (arithmetic operation, comparison, data movement)
  - each instruction takes constant time
  - realistic assumption on the size of the numbers (to represent n, it takes log n bits)

Asymptotic notation: Big-O

$O(g(n)) = \{f(n) \mid$ **there exist positive constants** c **and** n_0 **such that** $0 \leq f(n) \leq c\, g(n)$ **or all** $n \geq n_0\}$

Asymptotic notation: Big-O

$O(g(n)) = \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq f(n) \leq c\, g(n)$ or all $n \geq n_0\}$

We write $f(n) = O(g(n))$ if there exist constants $c > 0$, $n_0 > 0$ such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$.

Asymptotic notation: Big-O

$O(g(n)) = \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \le f(n) \le c\, g(n)$ or all $n \ge n_0\}$

We write $f(n) = O(g(n))$ if there exist constants $c > 0$, $n_0 > 0$ such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$.

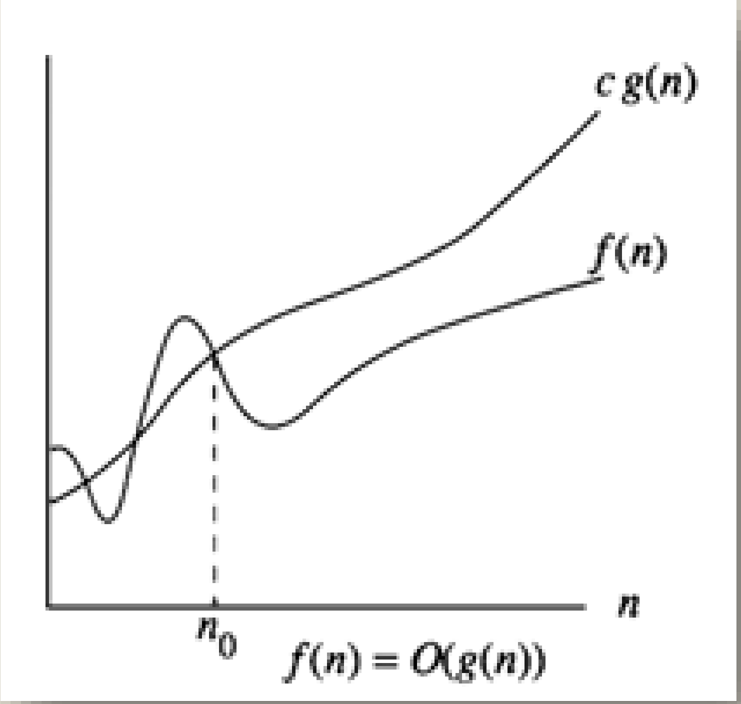
$4n + 23\log n - 28 = O(n)$

- Drops low-order terms
- Ignores leading constants
- May not hold for small values of n

$f(n) = O(g(n))$ if there exist constants $c > 0$, $n_0 > 0$ such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$.

$$
\begin{aligned}
f(n) &= 3n^2 - 4n + 512 \\
&\le 3n^2 + 512 \\
&\le 4n^2 \text{ for } n \ge 23
\end{aligned}
$$

- $f(n) = O(n^2)$
- $f(n) = O(n^3)$ also holds
- $f(n) = O(n)$ is false



CLRS, Figure 3.1

Which statements are true?

$3n^3 + 90n^2 – 5n = O(n^3)$

$3n^3 + 90n^2 – 5n = O(2^n)$

$3n^3 + 90n^2 – 5n = O(n^2)$

$5 \log n = O(n)$

$\sqrt{n} = O(\log n^8)$

$n \log n = O(n)$

$4n = O(n \log n)$

$n/\log n = O(\sqrt{n})$

Which statements are true?

$3n^3 + 90n^2 - 5n = O(n^3)$ true

$3n^3 + 90n^2 - 5n = O(2^n)$ true

$3n^3 + 90n^2 - 5n = O(n^2)$ false

$5 \log n = O(n)$ true

$\sqrt{n} = O(\log n^8)$ false

$n \log n = O(n)$ false

$4n = O(n \log n)$ true

$n/\log n = O(\sqrt{n})$ false

Asymptotic Bounds: Big Θ

$\mathrm{O}(g(n)) = \{ f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq f(n) \leq c\, g(n)$ for all $n \geq n_0 \}$

O captures upper bounds

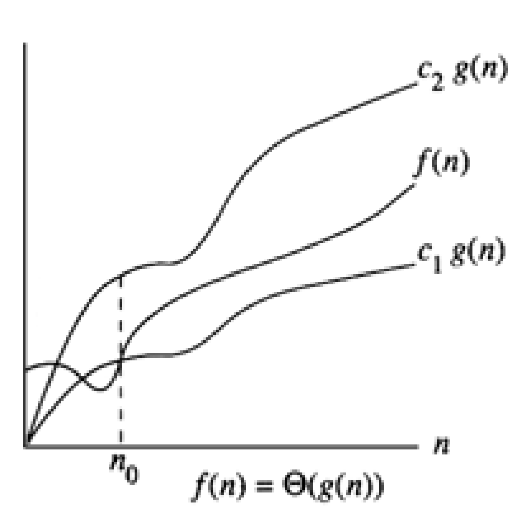
Asymptotic Bounds: Big Θ

$\mathrm{O}(g(n)) = \{ f(n) \mid$ there exist positive constants c and n_0 such that $0 \le f(n) \le c\, g(n)$ for all $n \ge n_0\}$

O captures upper bounds

$\Theta(g(n)) = \{ f(n) \mid$ there exist positive constants c_1, c_2, and n_0 such that $0 \le c_1\, g(n) \le f(n) \le c_2\, g(n)$ for all $n \ge n_0\}$

Θ captures upper and lower bounds



Examples

- $3n^3 + 90n^2 - 5n$ is $O(n^3)$
- $3n^3 + 90n^2 - 5n$ is $O(2^n)$
- 5 log n is O(n)
- 4n is O(n log n)

What about the Θ relationships?

Examples

- $3n^3 + 90n^2 - 5n = O(n^3)$ and $\Theta(n^3)$ is true
- $3n^3 + 90n^2 - 5n = O(2^n)$, but $\Theta(2^n)$ false
- 5 log n = O(n), but $\Theta(n)$ false
- 4n = O(n log n), but $\Theta(n \log n)$ false

Asymptotic Bounds: Big Ω

$O(g(n)) = \{ f(n) \mid$ there exist positive constants c and n_0
such that $0 \leq f(n) \leq c\, g(n)$ for all $n \geq n_0\}$

O captures upper bounds

$\Theta(g(n)) = \{ f(n) \mid$ there exist positive constants c_1, c_2, and n_0
such that $0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n)$ for all $n \geq n_0\}$

Θ captures upper and lower bounds

$\Omega(g(n)) = \{ f(n) \mid$ there exist positive constants c and n_0
such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n_0\}$

Ω captures lower bounds

$4n \log n = \Omega(n)$

Examples

- $32n^2 + 17n$ is…
- $32n^2 + 17n$ is…
- $32n^2 + 17n$ is…
- $32n^2 + 17n$ is…
- $6n^2 + 3n^{3/2}$ is…

Examples

- $32n^2 + 17n$ is $\Theta(n^2)$, $O(n^3)$, $\Omega(n)$
- $32n^2 + 17n$ is $\Omega(\log n)$, $\Omega(n \log n)$
- $32n^2 + 17n$ is not $\Omega(n^3)$
- $32n^2 + 17n$ is $\Omega(n^2)$ and $O(n^2)$ and thus also $\Theta(n^2)$
- $6n^2 + 3n^{3/2}$ is not $\Theta(n^{3/2})$, not $O(n^{3/2})$, $\Theta(n^2)$

Examples

- $32n^2 + 17n$ is $\Theta(n^2)$, $O(n^3)$, $\Omega(n)$
- $32n^2 + 17n$ is $\Omega(\log n)$, $\Omega(n \log n)$
- $32n^2 + 17n$ is not $\Omega(n^3)$
- $32n^2 + 17n$ is $\Omega(n^2)$ and $O(n^2)$ and thus also $\Theta(n^2)$
- $6n^2 + 3n^{3/2}$ is not $\Theta(n^{3/2})$, not $O(n^{3/2})$, $\Theta(n^2)$

$\sqrt{n} = \Omega(\log n)$

$n^{1/2} \geq c \log n$ (set $c=1$)

$\log n \geq 2 \log \log n$

$c=1$, $n_0 = 16$

Exercise: Show that $\mathbf{n^{3/2} + 6n^{3/4} + n \log n = \Theta(n^{3/2})}$

Exercise: Show that $\mathbf{n^{3/2} + 6n^{3/4} + n \log n = \Theta(n^{3/2})}$

i.e. $c_1 n^{3/2} \le n^{3/2} + 6n^{3/4} + n \log n \le c_2 n^{3/2}$ for some c_1, c_2, $n > n_0$

Exercise: Show that $\mathbf{n^{3/2} + 6n^{3/4} + n \log n = \Theta(n^{3/2})}$

i.e. $c_1 n^{3/2} \le n^{3/2} + 6n^{3/4} + n \log n \le c_2 n^{3/2}$ for some c_1, c_2, $n > n_0$

(1) $n^{3/2} + 6n^{3/4} + n \log n \le 7n^{3/2} + n \log n \le 8n^{3/2}$

show that $n \log n \le n^{3/2}$

$\log n \le n^{1/2}$

$\log \log n \le (\log n)/2$ true for all $n > 2^4$

(2) $c_1 n^{3/2} \le n^{3/2} + 6n^{3/4} + n \log n$

$n^{3/2} \le n^{3/2} + 6n^{3/4} + n \log n$

We get $c_1=1$, $c_2=8$, $n>16$

Note

- We will generally assume that n is “nice”
  - E.g., power of 2
  - We are not implementing the algorithms and only need to consider crucial the boundary/special cases
- When asked to design an efficient algorithm
  - sometimes you will be given a target asymptotic bound
  - other times you need to find the “best” one
- You can use known data structures
  - State how they are implemented and give time bounds of operations