Lecture 14 Notes

• **Goals for this week**
  – Big-O complexity (GT Section 4)
  – Solving recurrences (DT Section 4)
  – For quiz: Tuesday material
  – *Get ready for graphs, graph properties, and graph algorithms!* (GT Sections 1, 2, 3)

• **Feedback survey** (anonymous google doc)
  – Thanks for your responses!
  – 131 responses … your lowest quiz score will be dropped

• **“Midterm consolidation”** (73 participated)
  – Mean after capping at 100: 80.9
  – Standard deviation: 18.15
  – Updated MT scores will be posted by Thursday (plan = post today)
  – If you have low scores on both original MT and this MC, please stop by OHs or otherwise arrange to see me
Main Points From Last Time

• **D/Q Mergesort**
  – Recurrence: \( T(n) = 2T(n/2) + n \) = time to sort \( n \) elements
  – Solution: \( T(n) = n \cdot \log_2 n + n \) proved by mathematical induction
  – Proved by mathematical induction
    • Also intuitive from recursion tree: \((\log n \text{ levels}) \times (n \text{ time/level})\)
  – Other D/Q: matrix multiplication: \( T(n) = 8T(n/2) + n^2 \)
  – Other D/Q: long multiplication: \( T(n) = 4T(n/2) + 3n \)

• **Tower of Hanoi**
  – Recursion: \( T(n) = 2T(n-1) + 1 \) = time to move \( n \) disks
    • Two inequalities \( \leq \), \( \geq \) gave us the equality
  – Solution: \( T(n) = 2^n - 1 \) proved by mathematical induction
  – Other recursion: determinant: \( T(n) = n \cdot T(n-1) \)

• **Basketball**:
  #ways to score \( n \) pts
  \[ S(n) = S(n-1) + S(n-2) \]
  \( = (n+1)^{\text{st}} \) Fibonacci number
Long Multiplication in Pete’s Discussion

- Goal: multiply $1980 \times 2315$ using a D/Q approach
- Let $a = 1980$, $b = 2315$
- Let $a_L = 19$, $a_R = 80$, $b_L = 23$, $b_R = 15$ (split into left, right halves)

\[
\begin{align*}
\phantom{= aLbR aRbR} & + aLbL aRbL \\
= aLbL \ (aLbR + aRbL) \ aRbR
\end{align*}
\]
$1980 \times 2315 = ?$

\[
\begin{array}{c}
\begin{array}{c}
19 \\
23
\end{array} & \begin{array}{c}
80 \\
15
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
(19)(15) & \color{red}{(80)(15)}
\end{array} \\
\begin{array}{c}
(19)(23) & \color{purple}{(23)(80)}
\end{array}
\end{array}
\]

\[
\begin{array}{cccc}
\text{=} & 437 & 2125 & 1200 \\
\text{*}10^4 & \text{*}10^2 & \text{*}10^0
\end{array}
\]

\[
= 4370000 + 212500 + 1200 = 4583700 \quad (= 1980 \times 2315)
\]
Last Week: Basketball Before You Were Born

- No 3-point field goal
- Hypothetical game score: UCSD 75, UCLA 64
- Assuming no 3-pointers, in how many ways could UCSD have accumulated 75 points?

Notation:
- $S(n)$ \equiv \# ways to score n points

Small Cases:
- $S(0) = 1$
- $S(1) = 1$
- $S(2) = 2$ (2 or 1-1)
- $S(3) = 3$ (2-1 or 1-2 or 1-1-1)

Is this familiar?

$$S(n) = S(n-1) + S(n-2)$$

\#ways to reach n

$= \#ways to reach n-2$ (plus a field goal)

+ \#ways to reach n-1 (plus a free throw)

$\Rightarrow$ Fibonacci numbers!
Recurrence Relation

• S(n) = S(n-1) + S(n-2)

• What is S(75)?
  – S(0) = 1, S(1) = 1, S(2) = 2, S(3) = 3, S(4) = 5, S(5) = 8, …

• Fibonacci numbers F(n): 1, 1, 2, 3, 5, 8, …
  – S(75) = F(76) = 76th Fibonacci number
Choosing Between Solutions (Algorithms)

• Criteria:
  – Correctness
  – Time resources
  – Hardware resources
  – Simplicity, clarity (practical issues)

• Will need:
  – Size ("n", number of bits, …), Complexity measures
  – Notion of “basic” ("unit-cost") machine operation

Goal: Understand resource consumption (aka "complexity") as a function of problem "size"
Fibonacci Numbers

- \( \text{fib1}(n) \) if \( n < 2 \) then return \( n \)
  else return \( \text{fib1}(n-1) + \text{fib1}(n-2) \)

- Analysis: \( T(n) = 1 \) if \( n < 2 \); \( T(n) = T(n-1) + T(n-2) \) otherwise
  \( T(n) = F(n) \)

- Next two slides: this is \( \sim (1.64)^n \)

- \( F(n) \) i.e., closed-form solution to recurrence
  \( F(n) = F(n-1) + F(n-2) \)

- End up at "1"s
  \( T(n) \sim F(n) \)
Solving the Fibonacci Recurrence

DT Theorem 9, page DT-48

- **Notation:** \( F(n) = F(n-1) + F(n-2) \)

- **Guess:** try \( F(n) = a^n \) for some \( a \)
  \[
  a^n = a^{n-1} + a^{n-2} \quad \Rightarrow \quad a^2 = a + 1 \\
  \Rightarrow \quad a^2 - a - 1 = 0
  \]

Roots of quadratic: \( a_1 = (1 + \sqrt{5})/2 \); \( a_2 = (1 - \sqrt{5})/2 \)

**What's missing?**

- Missing "initial conditions"
  \[
  F(0) = 0, \quad F(1) = 1, \quad F(2) = 1
  \]
Guess: try $F(n) = a^n$ for some $a$

$a^n = a^{n-1} + a^{n-2} \Rightarrow a^2 = a + 1$

$a^2 - a - 1 = 0$

Roots of quadratic: $a_1 = (1 + \sqrt{5})/2; a_2 = (1 - \sqrt{5})/2$

Use all of the information

We know that $F(1) = 1; F(2) = 1$ (initial conditions)

Theorem 9: Homogeneous linear recurrence:
any linear combination of $(a_1)^n, (a_2)^n$ is a solution

Set up two equations in two unknowns:

$c_1 (a_1)^1 + c_2 (a_2)^1 = F(1) = 1 \; ; \; c_1 (a_1)^2 + c_2 (a_2)^2 = F(2) = 1$

$c_1 = 1 / \sqrt{5} , c_2 = -1 / \sqrt{5}$

$F(n) = c_1 (a_1)^n + c_2 (a_2)^n$
Fibonacci Numbers

• \text{fib2}(n) \quad f[1] = 1; \quad f[2] = 2;
  \quad \text{for } j = 3 \text{ to } n \text{ do }
  \quad f[j] = f[j - 1] + f[j - 2]

• Analysis: \quad T(n) = n
  \quad – Saving your work ("caching") can be useful!

• .... But... note that \#bits in F(n) is \sim 0.694n
  \quad – \#bits is linear in n because F(n) is exponential in n
  \quad – So, if we count bit operations, need quadratic number of
    bitwise additions to get F(n)

  * i.e., if we're more careful about resource accounting!

• \text{Fibonacci: Can we do "better"?}
Not Obvious, But Here Is A Shortcut…

- \( \text{fib3}(n) \)
  - Consider 2x2 matrix \( M: m_{11} = 0, m_{12} = 1, m_{21} = 1, m_{22} = 1 \)
  - Observe: \( \begin{bmatrix} F(k) & F(k+1) \end{bmatrix}^T = M \times \begin{bmatrix} F(k-1) & F(k) \end{bmatrix}^T \)
    \[
    \begin{bmatrix} F(n+1) & F(n+2) \end{bmatrix}^T = M^n \times \begin{bmatrix} F(1) & F(2) \end{bmatrix}^T = M^n \times \begin{bmatrix} 1 & 1 \end{bmatrix}^T
    \]

  \[
  M^{76} = M^{64} \times M^8 \times M^4
  \]

  \( \Rightarrow \text{fib3 uses “addition chains”} \)
Quantifying “Better”, “Worse”

• Resources used in computation often depend on a natural parameter, \( n \), of the input
  – search/sort list \# items \( x > y \)
  – matrix mult largest dim \( x \times y ; x + y \)
  – traverse tree \# nodes follow ptr

• Asymptotic Notation “as \( n \) grows large”
  \( f \in O(g) \) if \( \exists c > 0, N \) s.t. \( \forall n > N, f(n) \leq cg(n) \)
  e.g., \( 200n^2 \in O(2n^{2.5}) \)
  e.g., \( 2n + 20 \in O(n^2) \)

  “\( f \) grows no faster than \( g \)”

  – \( f \in \Omega(g) \) if \( g \in O(f) \)
  – \( f \in \Theta(g) \) if \( g \in O(f) \) and \( f \in O(g) \)
  – \([ f \in o(g) \iff \lim_{n \to \infty} f(n)/g(n) = 0 ]\)
Using “Big-Oh” Notation – Examples

• Definition: \( f(n) \) is **monotonically growing** (non-decreasing) if \( n_1 \geq n_2 \implies f(n_1) \geq f(n_2) \)

• **Fact:** For all constants \( c > 0, a > 1 \), and for all monotonically growing \( f(n) \), \( (f(n))^c \in O(a^{f(n)}) \)

• **Corollary (take \( f(n) = n \)):** \( \forall c > 0, a > 1, n^c \in O(a^n) \)
  – Any exponential in \( n \) grows faster than any polynomial in \( n \)

• **Corollary (take \( f(n) = \log_a n \)):** \( \forall c > 0, a > 1, (\log_a n)^c \in O(a^{\log_a n}) = O(n) \)
  – Any polynomial in \( \log n \) grows slower than \( n^{c'}, c'>0 \)

• **Exercise:** \( f \in O(s), g \in O(r) \implies f+g \in O(s+r) \)

• **Exercise:** \( f \in O(s), g \in O(r) \implies f\cdot g \in O(s\cdot r) \)
Theorem 9 (DT-48)

**Theorem 9**: Let \(a_0, a_1, \ldots, a_n\) be a sequence of numbers. Suppose there are constants \(b\) and \(c\) such that

\[a_n = ba_{n-1} + ca_{n-2}\]  

for \(n \geq 2\).

Let \(r_1\) and \(r_2\) be the roots of the

*characteristic equation* \(r^2 - br - c = 0\). Also: *characteristic polynomial*

If there are two distinct real roots, \(r_1, r_2\):

\[a_n = \alpha r_1^n + \beta r_2^n\]  

for \(n \geq 0\)

where: \(a_0 = \alpha + \beta\) and \(a_1 = r_1\alpha + r_2\beta\)

If there is one repeated real root, \(r\):

\[a_n = \alpha r^n + \beta nr^n\]  

for \(n \geq 0\)

where: \(a_0 = \alpha\) and \(a_1 = r\alpha + r\beta\)
Example 1

- Find the exact solution to the recurrence equation:
  \[ a_n = a_{n-1} + 2a_{n-2}, \text{ where } a_0 = 1 \text{ and } a_1 = 8 \]

Resume here on Thursday.

- Please review Example 1 (distinct roots \( r_1, r_2 \))
  and Example 2 (double root \( r \)) before class!

- We will start GT Sections 1-3 on Thursday.
Example 1

• Find the exact solution to the recurrence equation:
  \[ a_n = a_{n-1} + 2a_{n-2}, \text{ where } a_0 = 1 \text{ and } a_1 = 8 \]

• Recall: the characteristic equation for \( a_n = ba_{n-1} + ca_{n-2} \) is
  \[ r^2 - br - c = 0 \]

• Our characteristic equation is?
Example 1

• Find the exact solution to the recurrence equation:
  \[ a_n = a_{n-1} + 2a_{n-2}, \text{ where } a_0 = 1 \text{ and } a_1 = 8 \]

• Recall: the characteristic equation for \( a_n = ba_{n-1} + ca_{n-2} \) is
  \[ r^2 - br - c = 0 \]

• Our characteristic equation is?
  \[ r^2 - r - 2 = 0 \]
Example 1

• Find the exact solution to the recurrence equation:
  \[ a_n = a_{n-1} + 2a_{n-2}, \text{ where } a_0 = 1 \text{ and } a_1 = 8 \]

• Recall: the characteristic equation for \( a_n = ba_{n-1} + ca_{n-2} \) is
  \[ r^2 - br - c = 0 \]

• Our characteristic equation is?
  \[ r^2 - r - 2 = 0 \]

• Solving for the roots:
  \[ r^2 - r - 2 = 0 \]
  \( (r - 2)(r + 1) = 0 \)
  \[ r_1 = 2, \quad r_2 = -1 \]
Example 1 Continued

• Find the exact solution to the recurrence equation:
  \[ a_n = a_{n-1} + 2a_{n-2}, \text{ where } a_0 = 1 \text{ and } a_1 = 8 \]

• Theorem 9 tells us that if there are two real roots:
  \[ a_n = \alpha r_1^n + \beta r_2^n \text{ for } n \geq 0 \]
  where \( a_0 = \alpha + \beta \), \( a_1 = r_1\alpha + r_2\beta \)
  Since here \( r_1 = 2, r_2 = -1 \): \( a_n = \alpha 2^n + \beta n(-1)^n \)
Example 1 Continued

• Find the exact solution to the recurrence equation:
  \[ a_n = a_{n-1} + 2a_{n-2}, \text{ where } a_0 = 1 \text{ and } a_1 = 8 \]

• Theorem 9 tells us that if there are two real roots:
  \[ a_n = \alpha r_1^n + \beta r_2^n \text{ for } n \geq 0 \]
  where \( a_0 = \alpha + \beta, a_1 = r_1\alpha + r_2\beta \)
  Since here \( r_1 = 2, r_2 = -1 \): \( a_n = \alpha 2^n + \beta n(-1)^n \)

• Plugging into equations for \( a_0 \) and \( a_1 \), solving for \( \alpha \) and \( \beta \):
  \[ 1 = \alpha + \beta \]
  \[ 8 = 2\alpha - \beta \]
  Add: \( 9 = 3\alpha \), so \( \alpha = 3, \beta = -2 \)
Example 1 Continued

• Find the exact solution to the recurrence equation:
  \[ a_n = a_{n-1} + 2a_{n-2}, \text{ where } a_0 = 1 \text{ and } a_1 = 8 \]

• Theorem 9 tells us that if there are two real roots:
  \[ a_n = \alpha r_1^n + \beta r_2^n \text{ for } n \geq 0 \]
  where \( a_0 = \alpha + \beta, \ a_1 = r_1\alpha + r_2\beta \)
  Since here \( r_1 = 2, \ r_2 = -1 \): \( a_n = \alpha 2^n + \beta n(-1)^n \)

• Plugging into equations for \( a_0 \) and \( a_1 \), solving for \( \alpha \) and \( \beta \):
  \[
  1 = \alpha + \beta \\
  8 = 2\alpha - \beta
  \]
  Add: \( 9 = 3\alpha \), so \( \alpha = 3, \ \beta = -2 \)

• Putting it all together: \( a_n = 3(2^n) - 2(-1)^n \)
Proof of correctness by induction on $n$

• Trying to prove: $a_n = 3(2^n) - 2(-1)^n$
  Given: $a_n = a_{n-1} + 2a_{n-2}$, $a_0 = 1$, $a_1 = 8$
Proof of correctness by induction on $n$

- Trying to prove: $a_n = 3(2^n) - 2(-1)^n$
  
  Given: $a_n = a_{n-1} + 2a_{n-2}$, $a_0 = 1$, $a_1 = 8$

- Base case (does our equation work for $a_0$ and $a_1$?):
  
  $a_0 = 3(2^0) - 2(-1)^0 = 1$ ✔

  $a_1 = 3(2^1) - 2(-1)^1 = 8$ ✔
Proof of correctness by induction on $n$

- Trying to prove: $a_n = 3(2^n) - 2(-1)^n$
  Given: $a_n = a_{n-1} + 2a_{n-2}$, $a_0 = 1$, $a_1 = 8$

- Base case (does our equation work for $a_0$ and $a_1$?):
  
  $a_0 = 3(2^0) - 2(-1)^0 = 1$ ✔
  
  $a_1 = 3(2^1) - 2(-1)^1 = 8$ ✔

- Inductive step (does it work for $a_n$, when $a_n = a_{n-1} + 2a_{n-2}$?):

  $a_n = a_{n-1} + 2a_{n-2}$
  
  $= 3(2^{n-1}) - 2(-1)^{n-1} + 2(3(2^{n-2}) - 2(-1)^{n-2})$
  
  $= 6(2^{n-2}) - 4((-1)^{n-2}) + 3(2^{n-1}) - 2(-1)^{n-1}$
  
  $= 3(2^{n-1}) + 4((-1)^{n-1}) + 3(2^{n-1}) - 2(-1)^{n-1}$
  
  $= 6(2^{n-1}) + 2((-1)^{n-1})$
  
  $= 3(2^n) - 2((-1)^n)$ ✔
Example 2: Gambler’s Ruin

- A gambler repeatedly bets a flipped coin will come up heads.
  - If the coin is heads, the gambler wins $1.
  - If the coin is tails, the gambler loses $1.
  - If the gambler ever reaches $M he/she will stop.
Example 2: Gambler’s Ruin

- A gambler repeatedly bets a flipped coin will come up heads.
  - If the coin is heads, the gambler wins $1.
  - If the coin is tails, the gambler loses $1.
  - If the gambler ever reaches $M he/she will stop.

- Let $P_k$ = probability gambler loses all $k$ he/she has (= “ruin”)

\[
P_k = P(H)P(\text{ruin}|H) + P(T)P(\text{ruin}|T)
\]

\[
P_k = \frac{1}{2}P(\text{ruin}|H) + \frac{1}{2}P(\text{ruin}|T) \quad \text{(since we’re flipping a coin)}
\]

\[
P_k = \frac{1}{2}P(\text{ruin}|\text{win }$1$) + \frac{1}{2}P(\text{ruin}|\text{lose }$1$)
\]

\[
P_k = \frac{1}{2}P_{k+1} + \frac{1}{2}P_{k-1}
\]
Example 2: Gambler’s Ruin

• A gambler repeatedly bets a flipped coin will come up heads.
  – If the coin is heads, the gambler wins $1.
  – If the coin is tails, the gambler loses $1.
  – If the gambler ever reaches $M he/she will stop.

• Let $P_k =$ probability gambler loses all $k$ he/she has (“ruin”)
  
  $P_k = P(H) \cdot P(\text{ruin}|H) + P(T) \cdot P(\text{ruin}|T)$
  
  $P_k = \frac{1}{2} \cdot P(\text{ruin}|H) + \frac{1}{2} \cdot P(\text{ruin}|T)$ (since we’re flipping a coin)
  
  $P_k = \frac{1}{2} \cdot P(\text{ruin}|\text{win }$ $1) + \frac{1}{2} \cdot P(\text{ruin}|\text{lose }$ $1)$

  $P_k = \frac{1}{2} \cdot P_{k+1} + \frac{1}{2} \cdot P_{k-1}$

If we manipulate the equation we can then apply Theorem 9…

\[-\frac{1}{2}P_{k+1} = -P_k + \frac{1}{2}P_{k-1}\]

$P_{k+1} = 2P_k - P_{k-1}$

$P_k = 2P_{k-1} - P_{k-2}$
Example 2: Gambler’s Ruin continued

- We just learned: \( P_k = 2P_{k-1} - P_{k-2} \)

- Recall: the characteristic equation for \( a_n = ba_{n-1} + ca_{n-2} \) is
  \[ r^2 - br - c = 0 \]

- Here ours is:
  \[ r^2 - 2r + 1 = 0 \]
  \((r-1)(r-1) = 0\)
  So we have the repeated root, \( r = 1 \)
Example 2: Gambler’s Ruin continued

• We just learned: $P_k = 2P_{k-1} - P_{k-2}$

• Recall: the characteristic equation for $a_n = ba_{n-1} + ca_{n-2}$ is
  \[ r^2 - br - c = 0 \]

• Here ours is:
  \[ r^2 - 2r + 1 = 0 \]
  \[ (r-1)(r-1) = 0 \]
  So we have the repeated root, $r = 1$

• We know $P_0 = 1$ (if we start with $0$ we’re already ruined)
• We know $P_M = 0$ (if we start with $M$ we quit playing the game)
Example 2: Gambler’s Ruin continued

• We just learned: $P_k = 2P_{k-1} - P_{k-2}, \ P_0 = 1, \ P_M = 0$

• Theorem 9 tells us that if there is one repeated real root, r:
  
  $a_n = \alpha r^n + \beta nr^n$ for $n \geq 0$

  where: $a_0 = \alpha$ and $a_1 = r\alpha + r\beta$

• Here $r = 1$, so $a_n = \alpha + \beta n$, $\alpha = 1$ (we don’t have an $a_1$)
Example 2: Gambler’s Ruin continued

- We just learned: \( P_k = 2P_{k-1} - P_{k-2}, \ P_0 = 1, \ P_M = 0 \)

- Theorem 9 tells us that if there is one repeated real root, \( r \):
  \[ a_n = \alpha r^n + \beta nr^n \quad \text{for } n \geq 0 \]
  where: \( a_0 = \alpha \) and \( a_1 = r\alpha + r\beta \)

- Here \( r = 1 \), so \( a_n = \alpha + \beta n \Rightarrow \alpha = 1 \) (we don’t have an \( a_1 \))

- Plugging into equation for \( a_n \), solving for \( \beta \):
  \[ 0 = \alpha + \beta M \]
  \( \Rightarrow 0 = 1 + \beta M \)
  \( \Rightarrow \beta M = -1 \)
  \( \Rightarrow \beta = -(1/M) \)
Example 2: Gambler’s Ruin continued

- We just learned: \( P_k = 2P_{k-1} - P_{k-2}, P_0 = 1, P_M = 0 \)

- Theorem 9 tells us that if there is one repeated real root, \( r \):
  \[
  a_n = \alpha r^n + \beta n r^n \text{ for } n \geq 0
  \]
  \[
  \text{where: } a_0 = \alpha \text{ and } a_1 = r\alpha + r\beta
  \]

- Here \( r = 1 \), so \( a_n = \alpha + \beta n \), \( \alpha = 1 \) (we don’t have an \( a_1 \))

- Plugging into equation for \( a_n \), solving for \( \beta \):
  \[
  0 = \alpha + \beta M, 0 = 1 + \beta M, \beta M = -1, \beta = -(1/M)
  \]

- Plugging into \( a_n \) again: \( P_n = 1 - n/M \)
Example 2: Gambler’s Ruin continued

- $P_n = 1 - n/M$, so our probability of ruin, $P_k = 1 - k/M$
Example 2: Gambler’s Ruin continued

• \( P_n = 1 - n/M \), so our probability of ruin, \( P_k = 1 - k/M \)

• If we have $10 and won’t stop playing unless we have $100 what is the probability that we will lose all $10?

  \( k = 10 \)
  
  \( M = 100 \)
  
  \( P_{10} = 1 - 10/100 = .90 \), or 90%!
Example 2: Gambler’s Ruin continued

- \( P_n = 1 - \frac{n}{M} \), so our probability of ruin, \( P_k = 1 - \frac{k}{M} \)

- If we have $10 and won’t stop playing unless we have $100, what is the probability that we will lose our $10?
  
  \( k = 10 \)
  
  \( M = 100 \)
  
  \( P_{10} = 1 - \frac{10}{100} = .90 \), or 90%!

- If we have $10 but only want to win $12, what is the probability that we will lose our initial $10?
  
  \( P_{10} = 1 - \frac{10}{12} = \frac{1}{6} = 0.1667 \), now only 16.67%
Problems 14

- P14.1 Prove, using mathematical induction, that a $2^n \times 2^n$ chessboard that is missing any single $1 \times 1$ square can always be tiled by L-shaped triominoes. (A triomino is shown at right.)

- P14.2 A unit circle contains seven points. Prove that there exists a pair of points separated by distance $\leq 1$.

- P14.3 Same as P14.2, but with six points.
• Consider a recurrence of the form $a_n = ba_{n-1} + ca_{n-2}$ for $n \geq 2$

• Theorem 9 says that the form of the explicit solution depends on the roots of the characteristic polynomial $r^2 = br + c$

• If there are two distinct real roots $r_1$ and $r_2$, $a_n = \alpha r_1^n + \beta r_2^n$

• If there is one repeated real root $r$, $a_n = \alpha r^n + \beta nr^n$
Complexity of DQ for Long Multiplication

• This slide: illustrate “unrolling” of recurrence or, “substitution”
  – First line of attack if no convenient theorem available…

• Multiply two 2s-digit numbers, [wx] \cdot [yz]
  – 4 n/2-digit multiplications: xz, wz, xy, wy
  – Digit-shifting: multiplication by 10^s, 10^{2s}
  – 3 additions

\[ T(n) = 4T(n/2) + \theta(n) \]

• \[ T(n) \leq 4T(n/2) + cn \]
  \[ \leq 4 \left[ 4T(n/4) + cn/2 \right] + cn \]
  \[ = 16T(n/4) + (1 + 2)cn \]
  \[ \leq 16 \left[ 4T(n/8) + cn/4 \right] + (1 + 2)cn \]
  \[ = 64T(n/8) + (1 + 2 + 4)cn \]
  …
  \[ \leq 4^k T(n/2^k) + (1 + 2 + 4 + \ldots + 2^{k-1})cn \]

For \( k = \log_2 n \): \( T(n) \leq n^2 T(1) + cn^2 = O(n^2) \)

\( O(n^2) \) makes sense
Recurrence \( T(n) \leq a \cdot T(n/b) + O(n^d) \)

1) \( T(n) = O(n^d) \) if \( a < b^d \)

2) \( T(n) = O(n^d \log n) \) if \( a = b^d \)

3) \( T(n) = O(n^{\log_b a}) \) if \( a > b^d \)

Type (3): long multiplication, matrix multiplication

Type (2): mergesort
Master Theorem Examples

- Mergesort $T(n) = 2T(n/2) + \Theta(n)$
  
  $$T(n) = \Theta(n \log n)$$

- Matrix Multiply $T(n) = 8T(n/2) + \Theta(n^2)$
  
  $$T(n) = \Theta(n^{\log_2 8}) \approx \Theta(n^3)$$
Proof of “Master Theorem”

This is CSE 101 material: don’t worry for now, but you’ll need this in a year

Recurrence \( T(n) \leq a \cdot T(n/b) + O(n^d) \)

1) \( T(n) = O(n^d) \) if \( a < b^d \)

2) \( T(n) = O(n^d \log n) \) if \( a = b^d \)

3) \( T(n) = O(n^{\log_b a}) \) if \( a > b^d \)

• Assume \( n \) is a power of \( b \) \( \rightarrow \) can ignore rounding in \( \lceil n/b \rceil \)
• Subproblem size decreases by factor of \( b \) with each level of recursion \( \rightarrow \) reaches base case after \( \log_b n \) levels
• Branching factor \( a \) \( \rightarrow \) \( k \)th level of tree has \( a^k \) subproblems each of size \( n/b^k \)
• Total work done at \( k \)th level is \( a^k \times O(n/b^k)^d = O(n^d) \times \left(\frac{a}{b^d}\right)^k \)
Proof of “Master Theorem”
This is CSE 101 material: don’t worry for now, but you’ll need this in a year

Recurrence \( T(n) \leq a \cdot T(n/b) + O(n^d) \)

1) \( T(n) = O(n^d) \) if \( a < b^d \)

2) \( T(n) = O(n^d \log n) \) if \( a = b^d \)

3) \( T(n) = O(n^{\log_b a}) \) if \( a > b^d \)

- Total work done at \( k \)th level is \( a^k \cdot O(n/b^k)^d = O(n^d) \cdot (a / b^d)^k \)
- As \( k \) goes from 0 (root) to \( \log_b n \) (leaves), have geometric series with ratio \( a / b^d \)
  - \( a / b^d < 1 \rightarrow \) sum is \( O(n^d) \)
  - \( a / b^d > 1 \rightarrow \) sum is given by last term, \( O(n^{\log_b a}) \)
  - \( a / b^d = 1 \rightarrow \) sum is given by \( O(\log n) \) terms equal to \( O(n^d) \)
EXTRA