Warm Up
What is the asymptotic run time of MergeSort if its recurrence is

\[ T(n) = 2T\left(\frac{n}{2}\right) + 209n \]

Tree method

\[ T(n) = 2T\left(\frac{n}{2}\right) + 209n \]

Today's Keywords

- Karatsuba (finishing up)
- Guess and Check Method
- Induction
- Master Theorem
CLRS Readings

• Chapter 4

Homeworks

• Hw1 due Wed, January 30 at 11pm
  – Start early!
  – Written (use Latex) – Submit BOTH pdf and zip!
  – Asymptotic notation
  – Recurrences
  – Divide and Conquer

---

\[ \begin{array}{c}
\frac{a}{b} \times \frac{c}{d} \\
\text{Karatsuba Algorithm}
\end{array} \]

1. Recursively compute: \( ac, bd, (a + b)(c + d) \)
2. \((ad + bc) = (a + b)(c + d) - ac - bd \)
3. Return \( 10^n(a) + 10^7(ad + bc) + bd \)

Pseudo-code

1. \( x = \text{Karatsuba}(a,c) \)
2. \( y = \text{Karatsuba}(a,d) \)
3. \( z = \text{Karatsuba}(a+b,c+d)-x-y \)
4. Return \( 10^n x + 10^{n/2} z + y \)
3. Use asymptotic notation to simplify
\[ T(n) = 3T\left(\frac{n}{2}\right) + 8n \]
\[ T(n) = 8n \sum_{i=0}^{\log_2 n} \left(\frac{\delta}{2}\right)^i \]
\[ T(n) = 8n \left(\frac{\delta}{2}\log_2 n + \frac{\delta}{2} - 1\right) \]

Math, math, and more math…(on board, see lecture supplemental)

\[ T(n) = 24(n\log_2 n)^3 - 16n = \Theta(n\log_2 n)^3 \]
\[ = \Theta(n^{1.585}) \]
Recurrence Solving Techniques

- Tree
- ?✓ Guess/Check (induction)
- “Cookbook”
- Substitution

Induction (review)

Goal: \( \forall k, P(k) \) holds

Base case(s): \( P(1) \) holds

Hypothesis: \( \forall x \leq x_0, P(x) \) holds

Inductive step: \( P(x_0) \Rightarrow P(x_0 + 1) \)

Guess and Check Intuition

- To Prove: \( T(n) = \Theta(g(n)) \)
- Consider: \( g_*(n) = \Theta(g(n)) \)
- Goal: show \( \exists n_0 \) s.t. \( \forall n > n_0, T(n) \leq g_*(n) \)
  - (definition of big O)
- Technique: Induction
  - Base cases:
    - show \( T(1) \leq g_*(1), T(2) \leq g_*(2), \ldots \) for a small number of cases
  - Hypothesis:
    - \( T(n) \leq g_*(n) \)
  - Inductive step:
    - \( T(n + 1) \leq g_*(n + 1) \)
Karatsuba Guess and Check (Loose)

\[ T(n) = 3T(n/2) + 8n \]

Goal:

\[ T(n) \leq 3000 n^{1.6} = O(n^{1.6}) \]

Base cases:

\[ T(1) = 8 \leq 3000 \]

\[ T(2) = 3(1) + 16 = 20 \leq 3000 \cdot 2^{1.6} \]

. up to some small \( k \)

Hypothesis:

\[ \forall n \leq x_k, T(n) \leq 3000 n^{1.6} \]

Inductive step:

\[ T(x_k + 1) \leq 3000(x_k + 1)^{1.6} \]

Math, math, and more math... (see board, see lecture supplements)

Karatsuba Guess and Check (Loose)

\[ G=1 \quad T(x_k) \leq 3000 (x_k)^{1.6}, \quad T(n) = 3T(n/2) + 8n \]

\[ T(x_k^{1.6}) = 3 \left( \frac{x_k^{1.6}}{2} \right)^{2} + 8(x_k^{1.6}) \]

\[ \leq 3 \left( 3000 \left( \frac{x_k^{1.6}}{2} \right)^{1.6} \right) + 8(x_k^{1.6}) \]

\[ = \frac{3}{2} \left( 3000 \left( x_k^{1.6} \right)^{1.6} \right) + 8(x_k^{1.6}) \]

\[ \leq 0.997 \left( 3000 \left( x_k^{1.6} \right)^{1.6} \right) + 8(x_k^{1.6}) \]

\[ = 3000(x_k^{1.6}) + 8(0.997(x_k^{1.6})) \]

\[ = 3000(x_k^{1.6}) + 8(x_k^{1.6}) \]

\[ \leq 3000(x_k^{1.6}) + 8(x_k^{1.6}) \]

Karatsuba Guess and Check (Loose)

\[ T(n) \leq 3000 (n)^{1.6} \]

\[ T(n) \leq O(n^{1.6}) \]
Mergesort Guess and Check

\[ T(n) = 2T\left(\frac{n}{2}\right) + n \]

Goal: \[ T(n) \leq n \log_2 n = O(n \log_2 n) \]

Base cases:
- \( T(1) = 0 \)
- \( T(2) = 2 \leq 2 \log_2 2 \)
  .. up to some small \( k \)

Hypothesis: \( \forall n \leq x_k \ T(n) \leq n \log_2 n \)

Inductive step: \[ T(x_k + 1) \leq (x_k + 1) \log_2 (x_k + 1) \]

Math, math, and more math... (see board, see lecture supplemental)

---

Mergesort Guess and Check

\[ T(x_n) = 2T\left(\frac{x_n}{2}\right) + (x_n + 1) \leq 2(x_n + 1) \log_2 \frac{x_n}{2} + (x_n + 1) \]

Goal: \[ T(n) \leq n \log_2 n \]

Base cases:
- \( T(1) = 0 \)
- \( T(2) = 2 \leq 2 \log_2 2 \)
  .. up to some small \( k \)

Hypothesis: \( \forall x_n \ T(n) \leq n \log_2 n \)

Inductive step: \[ T(x_k + 1) \leq (x_k + 1) \log_2 (x_k + 1) \leq n \log_2 n \]

Math, math, and more math... (see board, see lecture supplemental)

---

Karatsuba Guess and Check

\[ T(n) = 3T\left(\frac{n}{3}\right) + 8n \]

Goal: \[ T(n) \leq 24n \log_2 n - 16n = O(n \log_2 n) \]

Base cases: by inspection, holds for small \( n \) (at home)

Hypothesis: \( \forall n \leq x_0 \ T(n) \leq 24n \log_2 n - 16n \)

Inductive step: \[ T(x_k + 1) \leq 24(x_k + 1) \log_2 (x_k + 1) - 16(x_k + 1) \]

Math, math, and more math... (see board, see lecture supplemental)
Karatsuba Guess and Check

\[ T(n) = 3T\left(\frac{n}{3}\right) + 16n \]

\[ T(n) = \Theta(n \log n) = O(n \log^2 n) \]

\[ T(n) = \Omega(n^\alpha) \]

**Base cases:**
\[ T(n) = \begin{cases} 2n^2 & \text{if } n = 1 \\ 2n & \text{if } n = 2 \end{cases} \]

**Hypothesis:**
\[ T(n) = 2n \log n \]

**Inductive step:**
\[ T(n) = 2n \log n + 2k \]

What if we leave out the $-16n$?

\[ T(n) = 3T\left(\frac{n}{3}\right) + 8n \]

**Goal:**
\[ T(n) = 24n \log_2 n + 518 = O(n \log^2 n) \]

**Base cases:**
\[ T(n) = \begin{cases} 0 & \text{if } n = 1 \\ 518 & \text{if } n = 2 \end{cases} \]

**Hypothesis:**
\[ T(n) = 2n \log_2 n \]

**Inductive step:**
\[ T(n) = 2n \log_2 n + 8k \]

"Bad Mergesort" Guess and Check

\[ T(n) = 2T\left(\frac{n}{2}\right) + 209n \]

**Goal:**
\[ T(n) = n \log_2 n + 209n \]

**Base cases:**
\[ T(1) = 0 \]
\[ T(2) = 518 = 209 \cdot 2 \log_2 2 \]

**Hypothesis:**
\[ T(n) = n \log_2 n \]

**Inductive step:**
\[ T(n) = 2n \log_2 (n+1) \]
Recurrence Solving Techniques

- Tree
- Guess/Check
- "Cookbook"
- Substitution

Observation

- **Divide**: $D(n)$ time,
- **Conquer**: recurse on small problems, size $s$
- **Combine**: $C(n)$ time
- **Recurrence**: $T(n) = D(n) + \sum T(s) + C(n)$

Many D&C recurrences are of form:

$T(n) = aT \left( \frac{n}{b} \right) + f(n)$

General

$T(n) = aT \left( \frac{n}{b} \right) + f(n)$

$T(n) = \sum_{i=0}^{\log_b n} a^i f \left( \frac{n}{b^i} \right)$
3 Cases

\[ T(n) = f(n) + aT \left( \frac{n}{a} \right) + a^2f \left( \frac{n}{a^2} \right) + \cdots + a^i f \left( \frac{n}{a^i} \right) \]

- **Case 1**: Most work happens at the leaves.
- **Case 2**: Work happens consistently throughout.
- **Case 3**: Most work happens at the top of tree.

**Master Theorem**

\[ T(n) = aT \left( \frac{n}{a} \right) + f(n) \]

- **Case 1**: if \( f(n) = \Theta \left( n \log_b a - \epsilon \right) \) for some constant \( \epsilon > 0 \), then \( T(n) = \Theta(n \log_b a) \).
- **Case 2**: if \( f(n) = \Theta(n \log_b a) \), then \( T(n) = \Theta(n \log_b a \log n) \).
- **Case 3**: if \( f(n) = \Omega \left( n \log_b a + \epsilon \right) \) for some constant \( \epsilon > 0 \), and if \( a f \left( \frac{n}{a} \right) \leq c f(n) \) for some constant \( c < 1 \) and all sufficiently large \( n \), then \( T(n) = \Theta(f(n)) \).