CS3000: Algorithms & Data
Jonathan Ullman

Jonathan Ullman

Lecture 3:

- Divide and Conquer: Karatsuba
- Solving Recurrences

Jan 13, 2020

Recap: Ask the Audience

- **"Big-Oh" Notation:** $f(n) = O\big(g(n)\big)$ if there exists $c \in (0, \infty)$ and $n_0 \in \mathbb{N}$ such that $f(n) \leq c \cdot g(n)$ for every $n \geq n_0$.

- Which of these statements are true?
  - $3n^2 + n = O(n^3)$
  - $n^3 = O(n^2)$
  - $3n^2 + n = O(n^2)$
  - $\log_2(n^2) = O(\log_2(n))$

Mergesort Wrapup

Integer Multiplication:
Karatsuba's Algorithm

Addition

- Given n-digit numbers u, v output $u + v$

	1	2	3	4
+	1	1	2	2
=	2	3	5	6

Multiplication

- Given n-digit numbers u, v output $u \cdot v$

				1	2	3	4
			x	1	1	2	2
				2	4	6	8
+			2	4	6	8	0
+		1	2	3	4	0	0
+	1	2	3	4	0	0	0
	1	3	8	4	5	4	8

Divide and Conquer Multiplication

	1	**2**	**3**	**4**
x	**1**	**1**	**2**	**2**

$$\boldsymbol{u} = 10^2 \cdot \mathbf{12} + \mathbf{34}$$

$$\boldsymbol{v} = 10^2 \cdot \mathbf{11} + \mathbf{22}$$

	$\boldsymbol{a}$	$\boldsymbol{b}$
x	$\boldsymbol{c}$	$\boldsymbol{d}$

$$\boldsymbol{u} = 10^{n/2} \cdot \boldsymbol{a} + \boldsymbol{b}$$

$$\boldsymbol{v} = 10^{n/2} \cdot \boldsymbol{c} + \boldsymbol{d}$$

Divide and Conquer Multiplication

	$\boldsymbol{a}$	$\boldsymbol{b}$
x	$\boldsymbol{c}$	$\boldsymbol{d}$

$$\boldsymbol{u} = 10^{n/2} \cdot \boldsymbol{a} + \boldsymbol{b}$$

$$\boldsymbol{v} = 10^{n/2} \cdot \boldsymbol{c} + \boldsymbol{d}$$

$$\boldsymbol{u} \cdot \boldsymbol{v} = (10^{n/2} \cdot \boldsymbol{a} + \boldsymbol{b})(10^{n/2} \cdot \boldsymbol{c} + \boldsymbol{d})$$

$$= 10^{n} \cdot \boldsymbol{ac} + 10^{n/2} \cdot (\boldsymbol{ad} + \boldsymbol{bc}) + \boldsymbol{bd}$$

- Four $n/2$-digit mults, three n-digit adds
  - Multiplying by 10^{n} is “free” because it’s a shift
- Recurrence: $T(n) = 4T\left(\frac{n}{2}\right) + 3n$

Divide and Conquer Multiplication

$$T(n) = 4 \cdot T(n/2) + 3n$$
$$T(1) = 1$$

- **Claim:** $T(n) \geq n^2$

Karatsuba's Algorithm

	a	b
x	c	d

$$\boldsymbol{u} = 10^{n/2} \cdot \boldsymbol{a} + \boldsymbol{b}$$

$$\boldsymbol{v} = 10^{n/2} \cdot \boldsymbol{c} + \boldsymbol{d}$$

$$\boldsymbol{u} \cdot \boldsymbol{v} = 10^{n} \cdot \boldsymbol{ac} + \ 10^{n/2} \cdot (\boldsymbol{ad} + \boldsymbol{bc}) + \boldsymbol{bd}$$

- Key Identity
  - $(\boldsymbol{b} - \boldsymbol{a})(\boldsymbol{c} - \boldsymbol{d}) = \boldsymbol{ad} + \boldsymbol{bc} - \boldsymbol{ac} - \boldsymbol{bd}$

- Only three $n/2$-digit mults (plus some adds)!

Karatsuba's Algorithm

```
Karatsuba(u,v,n):
  If (n = 1): Return u·v                      // Base Case

  Let m ← ⌈n/2⌉                               // Split
  Write u = 10^m · a + b,  v = 10^m · c + d

  Let e ← Karatsuba(a,c,m)                    // Recurse
      f ← Karatsuba(b,d,m)
      g ← Karatsuba(b-a,c-d,m)

  Return 10^(2m) · e + 10^m · (e + f + g) + f   // Merge
```

Correctness of Karatsuba

- **Claim:** The algorithm `Karatsuba` is correct

Running Time of Karatsuba

```
Karatsuba(u,v,n):
  If (n = 1): Return u·v

  Let m ← ⌈n/2⌉
  Write u = 10^m a + b,  v = 10^m c + d

  Let e ← Karatsuba(a,c,m)
      f ← Karatsuba(b,d,m)
      g ← Karatsuba(b-a,c-d,m)

  Return 10^{2m} e + 10^m (e + f + g) + f
```

Recursion Tree

$$T(n) = 3 \cdot T(n/2) + Cn$$
$$T(1) = C$$

Geometric Series

- Series $S = \sum_{i=0}^{\ell-1} r^i$

- Solution:
  - $r \neq 1,\ S = \frac{1-r^\ell}{1-r} = \frac{r^\ell-1}{r-1}$
  - $r = 1,\ S = \ell$

Karatsuba Wrapup

- Multiply n digit numbers in $O(n^{1.59})$ time
  - Improves over naïve $O(n^2)$ time algorithm
  - **Fast Fourier Transform:** multiply in $\approx O(n \log n)$ time
- Divide-and-conquer approach
  - Uses a clever algebraic trick to split
  - **Key Fact:** adding is faster than multiplying
- Prove correctness via induction
- Analyze running time via recursion tree
  - $T(n) = 3T(n/2) + Cn$

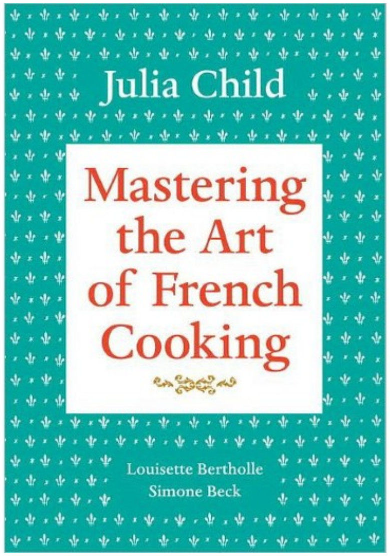
Ask the Audience!

- $T(n) = 3T(n/2) + n^2$
- $T(1) = 1$

Solving Recurrences:
“The Master Theorem”

The "Master Theorem"

- Generic divide-and-conquer algorithm:
  - Split into $\boldsymbol{a}$ pieces of size $\frac{\boldsymbol{n}}{\boldsymbol{b}}$ and merge in time $O\left(n^{\boldsymbol{d}}\right)$
- Recipe for recurrences of the form:
  - $T(n) = \boldsymbol{a} \cdot T(n/\boldsymbol{b}) + Cn^{\boldsymbol{d}}$



Recursion Tree

- $T(n) = \boldsymbol{a}T(n/\boldsymbol{b}) + n^{\boldsymbol{d}}$
- $\left(\frac{\boldsymbol{a}}{\boldsymbol{b}^{\boldsymbol{d}}}\right) > 1$

Recursion Tree

- $T(n) = \boldsymbol{a}T(n/\boldsymbol{b}) + n^{\boldsymbol{d}}$
- $\left(\frac{\boldsymbol{a}}{\boldsymbol{b}^{\boldsymbol{d}}}\right) = 1$

Recursion Tree

- $T(n) = \boldsymbol{a}T(n/\boldsymbol{b}) + n^{\boldsymbol{d}}$
- $\left(\frac{\boldsymbol{a}}{\boldsymbol{b}^{\boldsymbol{d}}}\right) < 1$

The "Master Theorem"

- Recipe for recurrences of the form:
  - $T(n) = \boldsymbol{a} \cdot T(n/\boldsymbol{b}) + Cn^{\boldsymbol{d}}$
- Three cases:
  - $\left(\frac{\boldsymbol{a}}{\boldsymbol{b}^d}\right) > 1: \; T(n) = \Theta\left(n^{\log_{\boldsymbol{b}} \boldsymbol{a}}\right)$
  - $\left(\frac{\boldsymbol{a}}{\boldsymbol{b}^d}\right) = 1: \; T(n) = \Theta\left(n^{\boldsymbol{d}} \log n\right)$
  - $\left(\frac{\boldsymbol{a}}{\boldsymbol{b}^d}\right) < 1: \; T(n) = \Theta\left(n^{\boldsymbol{d}}\right)$

Ask the Audience!

- Use the Master Theorem to Solve:
  - $T(n) = 16 \cdot T\left(\frac{n}{4}\right) + n^2$
  - $T(n) = 21 \cdot T\left(\frac{n}{5}\right) + n^2$
  - $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + 1$
  - $T(n) = 1 \cdot T\left(\frac{n}{2}\right) + 1$

The "Master Theorem"

- **Even More General:** all recurrences of the form
  - $T(n) = \boldsymbol{a} \cdot T(n/\boldsymbol{b}) + f(n)$
- Three cases:
  - $f(n) = O\left(n^{(\log_{\boldsymbol{b}} \boldsymbol{a}) - \varepsilon}\right)$:
    - $T(n) = \Theta\left(n^{\log_{\boldsymbol{b}} \boldsymbol{a}}\right)$
  - $f(n) = \Theta\left(n^{\log_{\boldsymbol{b}} \boldsymbol{a}}\right)$:
    - $T(n) = \Theta(f(n) \cdot \log n)$
  - $f(n) = \Omega\left(n^{(\log_{\boldsymbol{b}} a) + \varepsilon}\right)$ **AND** $\boldsymbol{a} f\left(\frac{n}{\boldsymbol{b}}\right) \leq C f(n)$ for $C < 1$
    - $T(n) = \Theta\left(f(n)\right)$