Announcements / Reminders

- Late homework due now

- Homework 2 due next week:
  - 5.3, 5.5
  - Algorithm Write-Up Guidelines reminder
  - Group names on the paper

---

Closest Pair Algorithm

```java
Closest-Pair(p1, ..., pn) {
    Compute separation line L such that half the points are on one side and half on the other side.
    δ1 = Closest-Pair(left half)
    δ2 = Closest-Pair(right half)
    δ = min(δ1, δ2)
    Delete all points further than δ from separation line L
    Sort remaining points by y-coordinate.
    Scan points in y-order and compare distance between each point and next 11 neighbors. If any of these distances is less than δ, update δ.
    return δ.
}
```

Closest of Points: Analysis

Running time:

- \( T(n) = 2T(n/2) + O(n \log n) \) implies \( T(n) = O(n \log^2 n) \)
- \( T(n) = 2T(n/2) + O(n \log n) \) implies \( T(n) = O(n \log n) \)

---

Complex Multiplication

Complex multiplication, \((a + bi)(c + di) = x + yi\).

Grade-school: \( x = ac - bd, y = bc + ad \)
- 4 multiplications, 2 additions

Q. Is it possible to do with fewer multiplications?
Complex Multiplication

Complex multiplication: $(a + bi)(c + di) = ac - bd + (ad + bc)i$.

Grade-school: $x = ac - bd, y = ad + bc$.

4 multiplications, 2 additions

Q: Is it possible to do with fewer multiplications?

A: Yes. (Gauss) $x = ac - bd, y = (a + b)(c + d) - ac - bd$.

3 multiplications, 5 additions

Remark. Improvement if no hardware multiply.

Integer Addition

Addition: Given two $n$-bit integers $a$ and $b$, compute $a + b$.

Grade-school: $\Theta(n)$ bit operations.

$$\begin{array}{c}
\phantom{+}11101101 \\
+ \phantom{+}11100101 \\
= \phantom{+} \underline{10001101}
\end{array}$$

Remark. Grade-school addition algorithm is optimal.

Integer Multiplication

Multiplication: Given two $n$-bit integers $a$ and $b$, compute $a \times b$.

Grade-school: $\Theta(n^2)$ bit operations.

Int. Multiplication:

$$\begin{array}{c}
11101101 \\
\times \phantom{+}11100101 \\
\hline
\phantom{+}11101101 \\
\phantom{+}00000000 \\
\phantom{+}11101101 \\
\phantom{+}11101101 \\
\phantom{+}11101101 \\
\hline
\underline{1101101001}
\end{array}$$

Q: Is grade-school multiplication algorithm optimal?

Divide-and-Conquer Multiplication: Warmup

To multiply two $n$-bit integers $a$ and $b$:

1. Multiply four $\frac{n}{2}$-bit integers, recursively.
2. Add and shift to obtain result.

$$\begin{array}{c}
a = 2^{n-1}a_1 + a_0 \\
b = 2^{n-1}b_1 + b_0 \\
a \times b = 2^{n-1}(a_1b_1 + a_0b_0) + a_0b_0 \\
\end{array}$$

Ex. $a = 1001101_2, b = 1110001_2$.

$$\begin{array}{c}
a \times b = 47 \times 49 \\
= 2287_2 \\
= 1000111001_2
\end{array}$$

Karatsuba Multiplication

To multiply two $n$-bit integers $a$ and $b$:

1. Add two $\frac{n}{2}$ bit integers.
2. Multiply three $\frac{n}{2}$-bit integers, recursively.
3. Add, subtract, and shift to obtain result:

$$\begin{array}{c}
a = 2^{n-1}a_1 + a_0 \\
b = 2^{n-1}b_1 + b_0 \\
a \times b = 2^{n-1}(a_1b_1 + a_0b_0) + a_0b_0 \\
\end{array}$$
Karatsuba Multiplication

To multiply two $n$-bit integers $a$ and $b$:
- Add two $\frac{n}{2}$-bit integers.
- Multiply three $\frac{n}{2}$-bit integers, recursively.
- Add, subtract, and shift to obtain result.

\[
\begin{align*}
   a &= 2^{n/2}a_L + a_R \\
   b &= 2^{n/2}b_L + b_R \\
   ab &= 2^n a_L b_R + 2^{n/2}(a_L b_R + b_L a_R) + a_R b_L \\
   &= 2^{n/2}(a_L + b_L)(a_R + b_R) - a_R b_L - a_L b_R + a_L b_R
\end{align*}
\]

**Theorem.** [Karatsuba-Ofman 1962] Can multiply two $n$-bit integers in $O(n^{\log_23})$ bit operations.

\[
T(n) = 3T\left(\frac{n}{2}\right) + O(n^{\log_23})
\]

Fast Integer Division Too (I)

**Integer division.** Given two $n$-bit (or less) integers $a$ and $i$, compute quotient $q = a/i$ and remainder $r = a \bmod i$.

**Fact.** Complexity of integer division is same as integer multiplication.

To compute quotient $q$:
- Approximate $r = i / q$ using Newton’s method: $r_q = 2r_q - r^2_q / 2$.
- After log $n$ iterations, either $q = \lfloor r \rfloor$ or $q = \lceil r \rceil$.

Matrix Multiplication

**Matrix multiplication.** Given two $n$-by-$n$ matrices $A$ and $B$, compute $C = AB$.

Grade-school. $O(n^3)$ arithmetic operations.

\[
C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}
\]

Grade-school dot product algorithm is optimal.

\[
\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \times \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n
\]

\[
\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\
               & \vdots & \ddots & \vdots \\
        a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \times \begin{bmatrix} b_1 \\
                                          \vdots \\
                                          \vdots \\
                                          b_n \end{bmatrix} = \begin{bmatrix} a_{11} b_1 + \cdots + a_{1n} b_n \\
                                              \vdots \\
                                              \vdots \\
                                              a_{n1} b_1 + \cdots + a_{nn} b_n \end{bmatrix}
\]

7/21/09
Block Matrix Multiplication

\[
\begin{bmatrix}
152 & 154 & 164 & 170 \\
856 & 836 & 844 & 974 \\
856 & 894 & 912 & 974 \\
1206 & 1262 & 1316 & 1370
\end{bmatrix}
\times
\begin{bmatrix}
6 & 1 & 2 & 3 \\
4 & 5 & 6 & 3 \\
8 & 9 & 10 & 11 \\
12 & 13 & 14 & 15
\end{bmatrix}
= 
\begin{bmatrix}
16 & 17 & 18 & 19 \\
20 & 21 & 22 & 23 \\
24 & 25 & 26 & 27 \\
28 & 29 & 30 & 31
\end{bmatrix}
\]

\[
C_{11} = A_{11} \times B_{11} + A_{12} \times B_{21}
\]

\[
C_{11} = \begin{bmatrix} 0 & 1 & 2 & 3 \end{bmatrix} \times \begin{bmatrix} 16 & 17 & 18 & 19 \\ 20 & 21 & 22 & 23 \\ 24 & 25 & 26 & 27 \\ 28 & 29 & 30 & 31 \end{bmatrix} = \begin{bmatrix} 152 & 156 \end{bmatrix}
\]

\[
C_{11} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 6 & 7 \\ 8 & 9 \end{bmatrix} \times \begin{bmatrix} 16 & 17 & 18 & 19 \\ 20 & 21 & 22 & 23 \\ 24 & 25 & 26 & 27 \\ 28 & 29 & 30 & 31 \end{bmatrix} = \begin{bmatrix} 152 & 156 \\ 544 & 526 \\ 1208 & 1262 \\ 1316 & 1370 \end{bmatrix}
\]