CSOR 4231

Analysis of Algorithms

Algorithms

Everywhere

- maps
- fedex
- biology, chemistry
- computer OS
- car engine
- space shuttle

Fast Computers

Good Interfaces

Fast Algorithms ←

Design & Analysis of Algs.

- theory
- proofs
- eye to practice/implementations

Matrix Multiplication

$$\begin{bmatrix} 3 & 1 & 1 \\ 2 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 6 \\ 2 & 0 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 6 & 20 \\ 5 & 18 \end{bmatrix}$$

Input A $n \times m$

B $m \times p$

Output C $n \times p$

$C = A \cdot B$

```
n   for i = 1 to n
      for j = 1 to p
p       C[i,j] = 0
        for k = 1 to m
m         C[i,j] += A[i,k] · B[k,j]
```

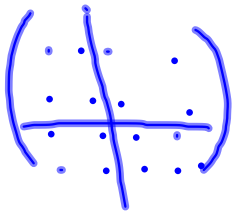
Running time: $O(n m p)$

$n = m = p$

$O(n^3)$

Lower bound $\Omega(n^2)$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & g \\ f & h \end{pmatrix} = \begin{pmatrix} r & s \\ t & u \end{pmatrix}$$



Use Recursion

$r = a \cdot e + b \cdot f$

$s = a \cdot g + b h$

$t = c e + d \cdot f$

$u = c \cdot g + d \cdot h$

Mult. 2 $n \times n$ matrices

$\Downarrow$

$\left(\frac{n}{2}\right)^3$ | 8 mult. $\frac{n}{2} \times \frac{n}{2}$ matr.

$\left(\frac{n}{2}\right)^2$ | 4 add $\frac{n}{2} \times \frac{n}{2}$ matr.

Let: $T(n)$ = time to mult. 2 $n \times n$ matrices

$$T(n) = \begin{cases} 8T\left(\frac{n}{2}\right) + 4\left(\frac{n}{2}\right)^2 & n > 1 \\ 1 & n = 1 \end{cases}$$

$= O(n^3)$

$T(n) = 7T\left(\frac{n}{2}\right) + 4n^2$

$O(n^{\log_2 7}) = O(n^{2.81})$

faster $O(n^{2.37\ldots})$