cse 102 5-9-24

Strassen's Algorithm (4.2 p. 75-83)

Problem: multiply 2 $n \times n$ sq. matrices

$$\underset{n \times n}{C} = \underset{n \times n}{A} \cdot \underset{n \times n}{B}$$

$$C = (c_{ij})_{1 \le i \le n,\ 1 \le j \le n}$$

where

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

Basic op. : multiplication of elements.

Runtime : $\Theta(n^3)$

Recursive solution

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right), \quad B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right)$$

$\frac{n}{2} \times \frac{n}{2}$

$$C_{11} = A_{11}B_{11} + A_{12}B_{21}$$

$$C_{12} = A_{11}B_{12} + A_{12}B_{22}$$

$$C_{21} = A_{21}B_{11} + A_{22}B_{21}$$

$$C_{22} = A_{21}B_{12} + A_{22}B_{22}$$

combine 4 $\frac{n}{2} \times \frac{n}{2}$ submatrices into

$$C = \left(\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right)$$

Exercise

- prove that this works!
- write pseudo-code.

note:

recursion to case $n = 1$ requires top level n is an exact power of 2.

let

$T(n) =$ # real multiplications on $n \times n$ arrays (n is a power of 2.)

observe

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

const. cost of dividing & combining. Book has n^2, since they count additions.

By case 1 of master theorem

$$T(n) = \Theta(n^3)$$

Volker Strassen: showed can compute $C_{11}, C_{12}, C_{21}, C_{22}$ using only 7 $\frac{n}{2} \times \frac{n}{2}$ matrix multiplications

Resulting in

$$T(n) = \begin{cases} 1 & n = 1 \\ 7T\left(\frac{n}{2}\right) + 1 & \end{cases}$$

By master thm

$$T(n) = \Theta\left(n^{\log_2 7}\right)$$

$$= \Theta\left(n^{2.8073\ldots}\right)$$

$$= o\left(n^3\right)$$

What if n is not an exact power of 2 ?

Augment A and B with 0 rows and 0 columns to A' and B' of size N, which is a power of 2

$$A' = \left(\begin{array}{c|c} A & 0 \\ \hline 0 & 0 \end{array}\right), \quad B' = \left(\begin{array}{c|c} B & 0 \\ \hline 0 & 0 \end{array}\right)$$

Exercise

let N the smallest power of 2 greater than or equal to n. show that

$$N^{\lg(7)} = \Theta\left(n^{\lg(7)}\right)$$

so augmentation has no (asymptotic) cost.

Dynamic Programming: Handout.

Ex Compute $\binom{n}{k}$

$$
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & \cdot & & \cdot & & 1 & & \\
 & 1 & & \cdot & & \cdot & & \cdot & & 1 & \\
1 & & \cdot & & \cdot & \underline{\underline{\binom{n}{k}}} & & \cdot & & & 1
\end{array}
$$

$\uparrow$

row = n
col = k