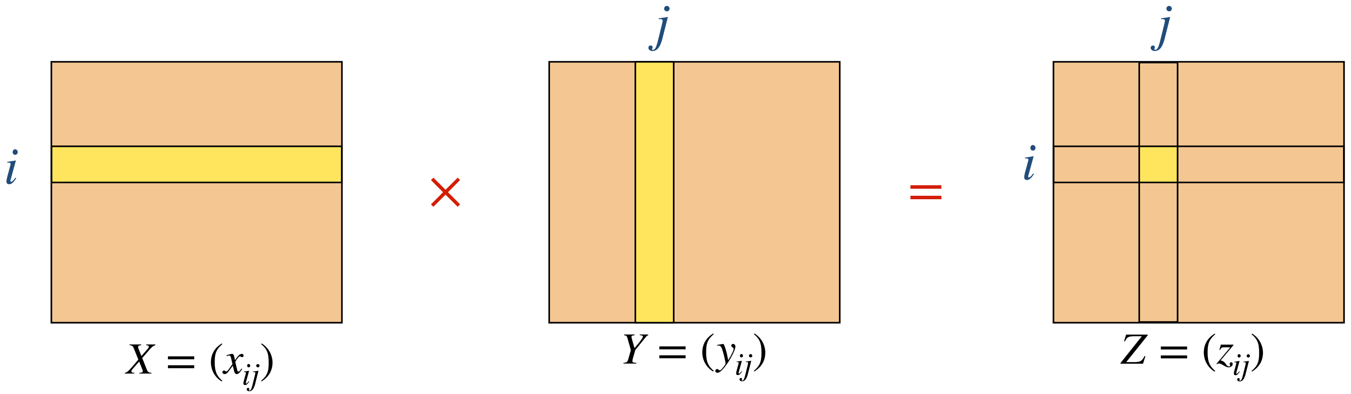
COL 351: Analysis and Design of Algorithms

Lecture 28

Matrix Multiplication
&
Transitive Closure

(Divide and Conquer strategy)

Matrix Multiplication of $n \times n$ Square Matrices



$$z_{ij} = \sum_{k=1}^{n} x_{ik} \, y_{kj}$$

Run time = $O(n^3)$

How efficiently can we multiply two matrices?

Transitive Closure

Given: Directed graph $G = (V, E)$.

Find: For each $v \in V$, the vertices reachable from v in graph G.

Naive approach: Run BFS / DFS algorithm from every vertex

- Run time: $O(mn) = O(n^3)$

How efficiently can we find transitive closure?

Product of 2x2 integer matrices

$$\underset{X}{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} \times \underset{Y}{\begin{bmatrix} e & f \\ g & h \end{bmatrix}} = \underset{Z}{\begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix}}$$

integer multiplications
= 8

integer additions
= 4

$z_{11} =$ $ae + bg$

$z_{12} =$ $af + bh$

$z_{21} =$ H. W.

$z_{22} =$ H. W.

Product of Block-matrices

$$\underbrace{\begin{bmatrix} [A] & [B] \\ [C] & [D] \end{bmatrix}}_{X} \times \underbrace{\begin{bmatrix} [E] & [F] \\ [G] & [H] \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} [Z_{11}] & [Z_{12}] \\ [Z_{21}] & [Z_{22}] \end{bmatrix}}_{Z}$$

Matrices

Ques

Z_{11} Z_{12} Z_{21} Z_{22} $\qquad$ $A \cdots H$

Can we find Z using product of smaller ($n/2$) matrices?

Product of Block-matrices

$$\begin{array}{cccccc} & & & j^{th}\ column & & j \\ i^{th}\ row & \begin{bmatrix} A & B \\ C & D \end{bmatrix} & \times & \begin{bmatrix} E & F \\ G & H \end{bmatrix} & = & i\ \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \end{array}$$

$Z_{11} =$ $AE + BG$ because we can prove that

$(Z_{11})_{ij} = (AE)_{ij} + (BG)_{ij}$

$Z_{12} =$ $AF + BH$

$Z_{21} =$ H.W.

$Z_{22} =$ H.W.

Recurrence relation in Simple Block-matrix-product is

$$T(n) = 8\ T(n/2) + d\ n^2$$

H. W.

$$T(n) = O(n^3)$$

Strassen's Algorithm

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$\begin{aligned} P_1 &= A(F - H) \\ P_2 &= (A + B)H \\ P_3 &= (C + D)E \\ P_4 &= D(G - E) \end{aligned} \qquad \begin{aligned} P_5 &= (A + D)(E + H) \\ P_6 &= (B - D)(G + H) \\ P_7 &= (A - C)(E + F) \end{aligned}$$

Let us compute $P_5 + P_4 - P_2 + P_6$

$A(E + H - H)$

$B(G + H - H)$

$D(E + G + H - E - G - H)$

Thus,

$P_5 + P_4 - P_2 + P_6$

$= AE + BG$

Recurrence relation in Strassen's Algorithm for matrix-product is
$T(n) \leq 7\ T(n/2) + d\ n^2$

$$\leq 7\left(7\,T(n/4) + d(n^2/4)\right) + dn^2$$

$$= 7^2\,T(n/4) + dn^2(1 + 7/4)$$

$$\leq 7^2\left(7\,T(n/8) + d(n/4)^2\right) + dn^2(1 + 7/4)$$

$$= 7^3\,T(n/8) + dn^2(1 + 7/4 + (7/4)^2)$$

$$\vdots$$

$$= 7^i\,T(n/2^i) + dn^2(1 + 7/4 + \cdots + (7/4)^{i-1})$$

$$= O(7^{\log_2 n} + (7/4)^{\log_2 n} \cdot dn^2)$$

$$= O(n^{\log_2 7} + n^{\log_2(7/4)+2}) = O(n^{\log_2 7}) = O(n^{2.81})$$

Homework Problem

Problem: What is complexity of squaring an $n \times n$ matrix in comparison to computing product of two $n \times n$ matrices?

State of Art: Matrix Multiplication

Strassen (1969) - $O(n^{2.81})$

$\vdots$

Coppersmith and Winograd (1990) - $O(n^{2.376})$

Alman and Williams (2020) - $O(n^{2.373})$

Notation "ω" is used to denote the "smallest" constant such that two $n \times n$ square matrices can be multiplies in $O(n^{\omega})$ time.

Transitive Closure

Given: Directed graph $G = (V, E)$.

Find: For each $v \in V$, the vertices reachable from v in graph G.

A is an adjacency-matrix of G

$$\text{if } A_{ij} = \begin{cases} 1 & \text{if } (i, j) \text{ is edge in } G \\ 0 & \text{otherwise} \end{cases}$$

T is an transitive-closure-matrix of G

$$\text{if } T_{ij} = \begin{cases} 1 & \text{there is a path from } (i) \text{ to } (j) \text{ in } G \\ 0 & \text{otherwise} \end{cases}$$

Lemma 1: Let A be adjacency matrix of a graph. Then,

$(A^k)_{ij} > 0$ iff there is a walk of length exactly 'k' from (i) to (j).

Part 1: If there exists a walk from i to j of size k, then $(A^k)_{ij} > 0$.

Let us suppose claim holds for $k-1$.

Consider a walk from i to j of size k. Let x = second last vertex.

$\Rightarrow$ i to x walk has size $k-1$

$\Rightarrow (A^{k-1})_{ix} > 0$, and also $A_{xj} = 1$

$\Rightarrow (A^k)_{ij} > 0$

Part 2: If $(A^k)_{ij} > 0$, then there exists a walk from i to j of size k.

Let us suppose claim holds for $k-1$.

If $(A^k)_{ij} > 0$ then $\exists$ an x s.t.

$(A^{k-1})_{ix} > 0$ and $A_{xj} > 0$

$\Rightarrow$ $\exists$ a walk from i to x of size $k-1$

and A_{xj} must be 1

$\Rightarrow (x, j)$ is edge

$\Rightarrow i \rightsquigarrow x \rightarrow j$ is a walk of size k.

Lemma 1: Let A be adjacency matrix of a graph. Then,

$(A^k)_{ij} > 0$ iff there is a walk of length exactly 'k' from (i) to (j).

Lemma 2: Let A be adjacency matrix of a graph. Then,

$((I + A)^k)_{ij} > 0$ iff there is a **walk of length at most 'k'** from (i) to (j).

Proof Sketch:

$$(I + A)^k = I + ({}^kC_1)A + ({}^kC_2)A^2 + \cdots + A^k$$

$(I + A^k)_{ij} > 0 \iff (A^q)_{ij} > 0$ for some $q \le k$.

$\iff \exists$ a walk of size q from i to j,
for some $q \le k$.

Transitive Closure

Transitive-Closure(A)

$M = I + A$;
$n = size(A)$;

For $(i = 1 \text{ to } \lceil \log_2 n \rceil)$ **do**:
  $M = M^2$
  Replace non-zero entries in M by 1;

Return M;

Transitive Closure

Transitive-Closure(A)

$M = I + A$;
$n = size(A)$;

For ($i = 1$ to $\lceil \log_2 n \rceil$) **do**:
    $M = M^2$
    Replace non-zero entries in M by 1;

Return M;

Theorem: The transitive closure of a directed graph with n vertices is computable in $O(n^{\omega} \log n)$ time.

The best known bound for transitive closure is: $O(n^{\omega})$