TAKE-HOME CLASS QUIZ: DUE MONDAY OCTOBER 28: MATRIX MULTIPLICATION AND INVERSION AS COMPUTATIONAL PROBLEMS

MATH 196, SECTION 57 (VIPUL NAIK)

Your name (print clearly in capital letters): _____

This quiz tests for a strong *conceptualization* (i.e., a metacognition) of the processes used for matrix multiplication and inversion. It is based on part of the `Matrix multiplication and inversion` notes and is related to Sections 2.3 and 2.4. It does not, however, test all aspects of that material.

PLEASE FEEL FREE TO DISCUSS *ALL* QUESTIONS.

(1) How many arithmetic operations are needed for naive matrix multiplication of a $m \times n$ matrix and a $n \times p$ matrix?
(A) $O(mnp)$ additions and $O(mnp)$ multiplications
(B) $O(m+n+p)$ additions and $O(mnp)$ multiplications
(C) $O(mn)$ additions and $O(np)$ multiplications
(D) $O(mn+mp)$ additions and $O(mnp)$ multiplications
(E) $O(m+n+p)$ additions and $O(m+n+p)$ multiplications

Your answer: _____

(2) What is the arithmetic complexity (in terms of total number of arithmetic operations needed) for naive matrix multiplication of two generic $n \times n$ matrices?
(A) $\Theta(n)$
(B) $\Theta(n^2)$
(C) $\Theta(n^3)$
(D) $\Theta(n^4)$
(E) $\Theta(n^5)$

Your answer: _____

(3) Which of the following is the tightest "obvious" lower bound on the possible arithmetic complexity of any generic algorithm for multiplying two $n \times n$ matrices? We use Ω to denote *at least that order*.
(A) $\Omega(n)$
(B) $\Omega(n^2)$
(C) $\Omega(n^3)$
(D) $\Omega(n^4)$
(E) $\Omega(n^5)$

Your answer: _____

(4) What is the minimum number of arithmetic operations needed to compute the product of two generic diagonal $n \times n$ matrices?
(A) n
(B) $n+1$
(C) $2n-1$
(D) $2n$
(E) n^2

Your answer: _____

(5) What is the minimum number of arithmetic operations needed to compute the product of a generic $1 \times n$ matrix and a generic $n \times 1$ matrix?

(A) n
(B) $n+1$
(C) $2n-1$
(D) $2n$
(E) n^2

Your answer: _____

(6) What is the minimum number of arithmetic operations needed to compute the product of a generic $n \times 1$ matrix and a generic $1 \times n$ matrix?
(A) n
(B) $n+1$
(C) $2n-1$
(D) $2n$
(E) n^2

Your answer: _____

(7) What is the minimum number of arithmetic operations needed to compute the product of a generic $n \times n$ diagonal matrix and a generic $n \times n$ upper triangular matrix? The upper triangular matrix has zero entries below the diagonal. The entries on or above the diagonal may be nonzero (and generically, they will be nonzero).
(A) $n(n-1)/2$
(B) $n(n+1)/2$
(C) $n(n-1)$
(D) n^2
(E) $n(n+1)$

Your answer: _____

Adding n numbers to each other requires $n-1$ addition operations. In a non-parallel setting, there is no way of improving this.

However, using the associativity of addition, we can write a faster parallelizable algorithm. A simple parallelization is to split the list being added into two sublists of length about $n/2$ each. Delegate the task of adding up within each sublist to different processors running in parallel. Then, add up the numbers obtained. This takes about half the time, with a little overhead (of collecting and adding up). This type of strategy is called a *divide and conquer* strategy. Using a divide and conquer strategy repeatedly, we can demonstrate that the parallelized arithmetic complexity of this approach is $\Theta(\log_2 n)$.

(8) Suppose A is a $1 \times n$ matrix and B is a $n \times 1$ matrix. Assume an unlimited number of processors that all have free read access to both A and B, free write access to the product matrix, and a shared workspace where they can store intermediate results. What is the arithmetic complexity in this context (i.e., the parallelized arithmetic complexity) for computing AB? What we mean here is: what is the smallest depth of a computational tree to compute AB?
(A) $\Theta(1)$
(B) $\Theta(\log_2 n)$
(C) $\Theta(n \log_2 n)$
(D) $\Theta(n^2)$
(E) $\Theta(n^2 \log_2 n)$

Your answer: _____

(9) Suppose A is a $n \times 1$ matrix and B is a $1 \times n$ matrix. Assume an unlimited number of processors that all have free read access to both A and B, free write access to the product matrix, and a shared workspace where they can store intermediate results. What is the arithmetic complexity in this

context (i.e., the parallelized arithmetic complexity) for computing AB? What we mean here is: what is the smallest depth of a computational tree to compute AB?
(A) $\Theta(1)$
(B) $\Theta(\log_2 n)$
(C) $\Theta(n \log_2 n)$
(D) $\Theta(n^2)$
(E) $\Theta(n^2 \log_2 n)$

Your answer: _____

(10) Suppose A and B are two $n \times n$ matrices. Assume an unlimited number of processors that all have free read access to both A and B, free write access to the product matrix, and a shared workspace where they can store intermediate results. What is the arithmetic complexity in this context (i.e., the parallelized arithmetic complexity) for computing AB? What we mean here is: what is the smallest depth of a computational tree to compute AB? Use naive matrix multiplication and speed it up using the parallelized processes discused here.
(A) $\Theta(1)$
(B) $\Theta(\log_2 n)$
(C) $\Theta(n \log_2 n)$
(D) $\Theta(n^2)$
(E) $\Theta(n^2 \log_2 n)$

Your answer: _____

We are given a $n \times n$ matrix A and we want to use *repeated squaring* to calculate powers of A. For instance, to calculate A^4, we can simply calculate $(A^2)^2$, which requires two multiplications. To calculate A^5, we calculate $(A^2)^2 A$, which requires three multiplications. Assume that we can store any number of intermediate matrices, i.e., storage space is not a constraint.

(11) What is the smallest number of matrix multiplications needed to calculate A^7 using repeated squaring?
(A) 3
(B) 4
(C) 5
(D) 6
(E) 7

Your answer: _____

(12) What is the smallest number of matrix multiplications needed to calculate A^8 using repeated squaring?
(A) 3
(B) 4
(C) 5
(D) 6
(E) 7

Your answer: _____

(13) What is the smallest number of matrix multiplications needed to calculate A^{21} using repeated squaring?
(A) 3
(B) 4
(C) 5
(D) 6
(E) 7

Your answer: _____

Suppose A is an *invertible* $n \times n$ matrix. It is possible to invert A using $\Theta(n^3)$ (worst-case) arithmetic operations via Gauss-Jordan elimination. We can thus add computation of the inverse to our toolkit when calculating powers. It is helpful even when calculating positive powers.

Count each matrix multiplication and each matrix inversion as one "matrix operation."

(14) What is the smallest positive r where we can achieve a saving on the total number of matrix operations to calculate A^r by also computing A^{-1}, rather than just using repeated squaring?

(A) 3
(B) 7
(C) 15
(D) 23
(E) 31

Your answer: _____

(15) *Strassen's algorithm* is a *fast matrix multiplication* algorithm that can multiply two $n \times n$ matrices using $O(n^{\log_2 7})$ arithmetic operations. In practice, however, a lot of existing computer code for matrix multiplication, written long after Strassen's algorithm was discovered, uses naive matrix multiplication. Which of the following reasons explain this? Please see Options (D) and (E) before answering.

(A) Strassen's algorithm becomes faster than naive matrix multiplication only for very large matrix sizes.
(B) Strassen's algorithm is more complicated to code.
(C) Strassen's algorithm is not as easily parallelizable as naive matrix multiplication.
(D) All of the above.
(E) None of the above.

Your answer: _____

There exist even faster algorithms for matrix multiplication than Strassen's algorithm. The best known algorithm currently is the *Coppersmith-Winograd algorithm*, which can multiply two $n \times n$ matrices in time $O(n^{2.3727})$. However, the Coppersmith-Winograd algorithm is even more rarely implemented than Strassen's for practical matrix multiplication code (according to some sources, Coppersmith-Winograd has *never* been implemented). The same reasons as those cited above for the reluctance to use Strassen's algorithm apply. There are some additional obstacles to practical implementations of the Coppersmith-Winograd algorithm that make it even more difficult to use.

(16) Suppose A and B are $n \times n$ matrices. What is the minimum number of matrix multiplications needed generically to compute the product $ABABABABA$?

(A) 4
(B) 5
(C) 6
(D) 7
(E) 8

Your answer: _____