Homework 19

Brian Knotten, Brett Schreiber, Brian Falkenstein

October 9, 2018

1

Here, we will reduce standard matrix multiplication to multiplication of two lower triangular matrices. Assume we have an algorithm $multTri(A, B)$ that solves the multiplication of two lower triangular matrices in $O(n^2)$ time. Given as input to standard matrix multiplication two matrices A and B, both of which are $n \times n$, we can construct two new matrices that will be passed into $multTri(A', B')$ by doing the following:

$$A' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & A & 0 \end{bmatrix}$$

$$B' = \begin{bmatrix} 0 & 0 & 0 \\ B & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Where 0 represents an $n \times n$ matrix of all zeros. Both of these matrices will be $3n \times 3n$. Both transformations take n^2 time, as it would take constant time to construct a $3n \times 3n$ matrix of zeros, then n^2 time to copy over the elements of A and B into the correct places in A' and B'. Multiplying A' and B' gives us the $3n \times 3n$ array:

$$A' \times B' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ AB & 0 & 0 \end{bmatrix}$$

We can then output the bottom left $n \times n$ elements in the matrix, which takes n^2 time. These elements are exactly the values of the elements of $A \times B$. Thus, we have converted the input from matrix multiplication to the input for $multTri(A, B)$ in n^2 time, and converted the output from $multTri(A, B)$ to the output for standard matrix multiplication in n^2 time, so if an $O(n^2)$ algorithm exists for the multiplication of two lower triangular matrices, so does one exist for standard matrix multiplication.

2

We will reduce standard matrix multiplication to the inversion of a nonsingular matrix. Assume we have an algorithm $invertMatr(C)$ that solves the inversion of a nonsingular matrix in $O(n^k)$, $k \geq 2$ time. Given two $n \times n$ matrices A and B as input to matrix multiplication, we can construct a new matrix to be passed into $invertMatr(C)$ as follows:

$$C = \begin{bmatrix} I & A & 0 \\ 0 & I & B \\ 0 & 0 & I \end{bmatrix}$$

Where 0 represents an $n \times n$ matrix of all zeros and I represents the Identity matrix with ones on the main diagonal and zeros everywhere else. C will be $3n \times 3n$. The construction of C will take approximately n^2 time, as it takes constant time to construct a $3n \times 3n$ identity matrix, then n^2 time to copy the elements of A and B over to the correct places in C. Inverting C gives us the $3n \times 3n$ matrix:

$$C^{-1} = \begin{bmatrix} I & -A & AB \\ 0 & I & -B \\ 0 & 0 & I \end{bmatrix}$$

We can output the top right $n \times n$ elements in the matrix C^{-1}, which are exactly the values of the elements in $A \times B$, in n^2 time. Thus, we have converted the input from the matrix multiplication of two $n \times n$ matrices to the input for $invertMatr(C)$ in n^2 time and converted the output from $invertMatr(C)$ to the output from matrix multiplication in n^2 time. Therefore, if an $O(n^k)$, $k \geq 2$ algorithm exists for the inversion of a nonsingular matrix, then an $O(n^k)$ algorithm also exists for the matrix multiplication of two $n \times n$ matrices.