CS 530 - Fall 2019
Homework 1 - Brief Answers

Due: Thursday, September 12 - to be submitted via Gradescope

Reading : Section 4.2, pages 75-82, where you will find Strassen's algorithm, and Sections 28.1 and 28.2, pages 813-830 of the textbook.

Review: See appendix D of the textbook for a review of matrices including definitions of different types of matrices and properties of matrix operations.

Also, scan through the first 75 pages of the first part of the book. Review what you need to and read any parts that are new.

Problems: Please limit your answer to the following problems to at most 2 pages each.

1. LU Matrices

Take a look at the definitions of lower triangular matrices (L), unit lower triangular matrices (L) and upper triangular matrices (U) in Appendix D, part 1 (Page 1217) of the textbook.

(i). What is the determinant of a lower triangular matrix ? Briefly justify your answer.

Answer: As with any triangular matrix the determinant is the product of the diagonal elements of the matrix.

Justification: As the matrix is lower triangular matrices, the only permutation which yields a non-zero contribution is the identity permutation, and the identity permeation correspond to the product of the diagonal elements. (You should convince yourself of this.)

(ii). Give a simple algorithm which will compute the inverse of a non-singular lower triangular matrix. Your algorithm should work in O(n) steps, or at least it should compute all of the non-zero elements of the inverse in time O(n). Explain briefly why your algorithm is correct.

Answer: This problem was removed from HW 1.

(iii). Is it possible for a singular lower triangular matrix to have an inverse ? Here either give an example showing this statement is true or prove that no singular lower triangular matrix has an inverse.

Answer: No. You can take the definition of singular to be "has no inverse", or start from a more intuitive def. of singular and prove it cannot be inverted. (Any short, reasonable justification is OK.)

(iv). Assume you are given a matrix M and a decomposition M=LU with L and U as above. Explain how to use this decomposition to compute M's determinant. The computation should be use your knowledge of L and U.

Answer: The determinant of the product of 2 matrices is the product of the two determinants. So in this case the determinant of M is det(L) $\times$ det(U), and det(L)=1 while det(U) is directly computable as in part i above.

2. LU Decomposition

(i). Prove that for n equal to 2 or 3 there is a non-singular square (n by n) matrix which has no LU decomposition with L unit lower triangular and U upper triangular. (In fact, this is true for any integer ≥ 2.)

Answer: For n = 2 consider

```
0 1
1 0
```

For n = 3 consider

```
 0 0 1
 0 1 0
 1 0 0
```

For the proofs try obtaining a contradiction from the assumption that the matrix can be decomposed as LU. (It gets a bit more difficult when you try to prove this for all n.)

(ii). We will see that all non-singular square matrices do have an LUP decomposition (some time soon in class). Here P is a permutation matrix, also defined in Appendix D and used in Chapter 28.

Show that the inverse of a permutation matrix P is also a permutation matrix. (You can do this by explicitly defining P^{-1} when P is given.)

Answer: If P is a permutation matrix then the inverse of P is P's transpose P^t. That is, we just interchange the rows and columns of P. But doing this preserves the definition of permutation matrix, namely that each row and each column contains exactly one 1.

(iii). Give an example of a singular 2 by 2 matrix A and of a singular 3 by 3 matrix B which have an LU decomposition. Both A and B should be non-zero matrices and you should write the L and the U for both of them.

Answer:

```
Let A = 0 -1      L = 1 0    U = 0 -1
        0 -2          4 1        0 -2


Let B = 0 -1 1    L = 1 0 0    U =0 -1 1
```

```
0 -2 7        4 1 0       0  2 3
0  6 4       -2 2 1       0  0 0
```

3. Strassen's method

Assume you are handed Strassen's algorithm which multiplies two 2 by 2 matrices and uses 7 integer multiplications and 18 integer additions to do this.

i. Show how to use this algorithm iteratively (using divide and conquer) to multiply two 4 by 4 matrices. Calculate exactly how many integer multiplications and integer additions are done? You should show your work and should compare the answer you get to the number of multiplications and adds which would be done if you turn the 4 by 4 algorithm into block multiplication of 2 by 2 matrices.

Answer: If we do the 4 by 4 matrix multiplication we use $4^3 = 64$ multiplications and $4^2 3 = 48$ additions.

Strassen, on the other hand, takes 7 multiplications and 18 +'s to multiply two 2×2 matrices. So to multiply two 4×4 we use 7x7 = 49 mults and 40+126+32 = 198 +'s.
The work it justify all these numbers is not shown here. However, as a hint the number of additions for Strassen's algorithm is 198 = 40+126+32 where 40 = 10×4, 126 = 7×16 and 32 = 8×4..

ii. In Section 4.2 of the textbook it is shown that Strassen's method uses only O($n^{log 7}$) multiplications when n is an even power of 2. (You need not prove this.)

Using this result describe how you can get the same O($n^{log 7}$) multiplications even if you start with n×n matrices with n not an even power of 2.

Answer: One idea is to let n' = 2^k, k an integer, be the smallest power of 2 bigger than n. Now simply pad (expand) the original matrices you are multiplying with 0's so that they become n'×n' matrices. While n' is larger that n, note that n' is at most 2n.

So now we are doing Strassen's algorithm for n'×n' matrices as usual and we know this takes O($n'^{log 7}$) multiplications. But as n' ≤ 2n we have at most O($(2n)^{log 7}$) multiplications which is still O($n^{log 7}$).

Furthermore the block multiplication on the n' by n' matrix works just as is did for the normal recursive Strassen algorithm since n' is a power of 2.