Spring 2017

CS3230 Tutorial (Divide and Conquer Algorithms, week of 13 February)

1. Let *A[0..n − 1]* be an array of *n* distinct numbers. An inversion is a pair of array elements (*A[i],A[j]*) with *i* < *j* but *A[i]* > *A[j]*. Design an *O*(*n* log *n*) algorithm to count the number of inversions in the array *A*, and prove the efficiency class of your algorithm.

2. A tromino is a 2×2 board with a missing square. Prove that a $2^n \times 2^n$ ($n \geqslant 1$) board with a square missing anywhere, can be completely covered by trominoes. Design an algorithm to cover a $2^n \times 2^n$ board with a missing square with trominoes.

3. Give exact solutions (with proof) for the following recurrences.

$$A(n) = \begin{cases} 0 & n = 1 \\ A(\lfloor n/2 \rfloor) + A(\lceil n/2 \rceil) + 1 & n > 1 \end{cases}$$

$$B(n) = \begin{cases} 1 & n = 1 \\ B(\lfloor n/2 \rfloor) + 1 & n > 1 \end{cases}$$

4. At most how many questions do you have to ask in order to find out the birth date of your girl (boy) friend? Assume she (he) will answer your questions honestly but with only "yes" or "no".

5. Consider the task of breaking an *m*×*n* chocolate bar into *mn* 1×1 pieces. How many breaks are needed?

6. Can a binary tree be recovered from its pre-order, in-order, post-order traversal lists?

7. Illustrate Strassen's algorithm by multiplying the following two 4 × 4 matrices.

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{bmatrix}$$