Tutorial 2, Design and Analysis of Algorithms, 2019

1. Show how to multiply the complex numbers $a + bi$ and $c + di$ using only three multiplications of real numbers. The algorithm should take a, b, c, and d as input and produce the real component $ac - bd$ and the imaginary component $ad + bc$. separately.

2. How would you modify Strassen's algorithm to multiply $n \times n$ matrices in which n is not an exact power of 2? Show that the resulting algorithm runs in time $\Theta(n^{\log_2 7})$.

3. What is the largest k such that if you can multiply 3×3 matrices using k multiplications (not assuming commutativity of multiplication), then you can multiply $n \times n$ matrices in time $o(n^{\log_2 7})$? What would the running time of this algorithm be?

4. How quickly can you multiply a $kn \times n$ matrix by an $n \times kn$ matrix, using Strassens algorithm as a subroutine? Answer the same question with the order of the input matrices reversed.

5. Professor F. Lake tells his class that it is asymptotically faster to square an $n-$bit integer than to multiply two $n-$bits integers. Should they belive him? Give reasons for your answer.

6. You are interested in analyzing some hard-to-obtain data from two separate databases. Each database contains n numerical values - so there are $2n$ values total - and you may assume that no two values are the same. You would like to determine the median of this set of $2n$ values, which we will define here to be the n'th smallest value. However, the only way you can access these values is through queries to the databases. In a single query, you can specify a value k to one of the two databases, and the chosen database will return the k'th smallest value that it contains. Since queries are expensive, you would like to compute the median using as few queries as possible. Give an algorithm that finds the median value using at most $O(\log n)$ queries.

7. Consider an n-node complete binary tree T, where $n = 2^d - 1$ for some d. Each node v of T is labeled with a real number x_v. You may assume that the real numbers labeling the nodes are all distinct. A node v of T is a *local minimum* if the label x_v is less than the label x_w for all nodes w that are joined to v by an edge. You are given such a complete binary tree T, but the labeling is only specified in the following *implicit* way: for each node v, you can determine the value x_v by probing the node v. Show how to find a local minimum of T using only $O(\log n)$ *probes* to the nodes of T. Give a proof of correctness of your algorithm and also prove its time complexity.