Tutorial 1, Design and Analysis of Algorithms, 2020

1. For a real number n the function $\log^*(n)$ is defined as follows: $\log^*(n)$ is the smallest natural number i so that after applying logarithm function (base 2) i times on n we get a number less than or equal to 1. E.g. $\log^*(2^2)$ is 2 because $\log(\log(2^2)) = 1 \leq 1$. $\log^*(2^{2^2})$ is 3 because $\log\log(\log(2^{2^2})) = 1 \leq 1$.
   Either prove or disprove:
   (a) $\log(\log^*(n)) = O(\log^*(\log(n)))$.
   (b) $\log^*(\log(n)) = O(\log(\log^*(n)))$.

2. Solve the following recurrence relation (without using the Master Theorem):

$$T(n) = \begin{cases} 1, & \text{for } n \leq 4; \\ 2T(\sqrt{n}) + \frac{\log n}{\log\log n}, & \text{for } n > 4. \end{cases}$$

3. In an infinite array, the first n cells contain integers in sorted order and the rest of the cells are filled with ∞. Present an algorithm that takes x as input and finds the position of x in the array in $O(\log n)$ time. You are not given the value of n.

4. Device a "binary" search algorithm that splits the set not into two sets of (almost) equal sizes but into two sets, one of which is twice the size of the other. How does this algorithm compare with binary search?

5. Device a ternary search algorithm that first tests the element at position $\frac{n}{3}$ for equality with some value x, and then checks the element at $\frac{2n}{3}$ and either discovers x or reduces the set size to one-third the size of the original. Compare this with binary search.

6. 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.

7. 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 $\Theta(n^{\log_2 7})$.

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

9. 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.

10. 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.