Tutorial 1, Design and Analysis of Algorithms, 2019

1. Let F_n denote the n^{th} Fibonacci number ($F_0 = F_1 = 1, F_n = F_{n-1}+F_{n-2}$ for $n \geq 2$). Prove that

   (a) $F_n = \Omega(2^{\frac{n}{2}})$.

   (b) $F_n = O(2^n)$.

   (c) $F_n = \Theta\left(\left(\frac{\sqrt{5}+1}{2}\right)^n\right)$.

2. 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) $\log^*(\log(n)) = O(\log(\log^*(n)))$

3. Solve the following recurrence relations (without using the Master Theorem):

   (a) $T(n) = 1$, for $n \leq 4$
   $T(n) = T(\sqrt{n}) + c$, for $n > 4$.

   (b) $T(n) = 1$, for $n \leq 4$
   $T(n) = 2T(\sqrt{n}) + \log n$, for $n > 4$.

4. Solve the following recurrence relation (without using the Master Theorem):
   $T(n) = 1$, for $n \leq 4$
   $T(n) = 2T(\sqrt{n}) + \frac{\log n}{\log\log n}$, for $n > 4$

5. 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.

6. 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?

7. 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.