Problem 1

Suppose you own \( n \) electrical devices. Each of them comes with a charger cable, which you tossed into a box when you got it. But now it is time to recharge the devices, and so you must find for each one the correct charger cable. For each device, exactly one charging cable is correct. The charging cables look similar enough that you cannot compare them amongst themselves. The only thing that you can do is plug a cable into a device, which will tell you whether the plug fits, or is too big, or is too small.

Argue that any algorithm to find cables for all devices must use \( \Omega(n \log n) \) such operations in the worst case.

Problem 2

Given an array \( A[0...n-1] \) of numbers, show that if \( A[i] \geq A[j] \) for all \( j \leq i \log n \), the array can be sorted in \( O(n \log \log n) \) time.

Problem 3

Let \( T \) be an AVL tree with \( n \) nodes and height \( h \). If \( N(h) \) is the minimal number of nodes \( T \) can have, then show that \( N(h) \geq 2^{\frac{h}{2}} \).

Problem 4

We have an array \( A \) of \( n \) non-negative integers such that each integer is less than \( k \). Give an algorithm with \( O(n + k) \) preprocessing time such that queries of the form “how many integers are there in \( A \) that are in the range \([a, b]\)?” can be answered in \( O(1) \) time.

Note that \( a \) and \( b \) are not fixed; they are parameters given to the query algorithm.

Problem 5

Prove that “Lazy Deletion” as described in class, for binary search trees is amortised \( O(search) \) time.