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.

Give a randomized algorithm that finds for all devices the matching charger cable and that uses expected $O(n \log n)$ operations of “try to plug into the device”.

Problem 2

Design an algorithm for the following problem:

Given array $A[0...n - 1]$ (pairwise distinct) elements and a number $k \in \{0, ..., n - 1\}$, rearrange the elements so that the first $k$ positions contain the $k$ smallest elements in sorted order.

a) Design an algorithm with $O(n + k \log k)$ expected time.

b) Now give an algorithm with worst case running time in $O(n + k \log n)$ and $O(1)$ extra space.

c) Give an algorithm with worst case running time in $O(n + k \log k)$.
Problem 3

Define a Cherry-tree to be a perfect binary tree with a cherry on top. The root has exactly one child, every other internal node has two children, and all leaves are on the same level. Here’s an example of a Cherry-tree:

```
          a
         /|
        / |\n       b  c  d
      / |  / |  / |
     e  f  g  h
```

Discuss how to merge two Cherry-trees of the same height in $O(1)$ time so that the output is a Cherry-tree.

Observe that a Cherry-tree is what we would get if we encoded a binomial tree (a tree in a binomial heap) as a binary tree.