- This problemset has *three* questions. The programming problems may be harder, or require more time, than their point value suggests.
- Please send the solutions to `gawry1+EDSCourse2014@gmail.com`
- The deadline is 22.06.2014 anywhere on Earth.

(20) 1. Prove the following statement (by induction on the number of nodes): if T is any rooted tree, and $d(v)$ denotes the degree (i.e., the number of children) of a node v, then $\sum_{v \in T} \max(0, d(v) - 1) = \|T\| - 1$, where $\|T\|$ is the number of leaves of T.

(60) 2. So far we were interested in answering a *single* predecessor query efficiently. But say that we get k such queries at once, and furthermore the queries are sorted, i.e., we get $x_1 < x_2 < \ldots x_k$. We call this a *batched predecessor query*. As usual, we assume that the stored sequence is also sorted, i.e., $a_1 < a_2 < \ldots a_n$.

(a) Show that if $k = n$, we can process such batched predecessor query in $\mathcal{O}(k)$ time.

(b) Show that we can process any batched predecessor query in $\mathcal{O}(k \log(\frac{2n}{k}))$ time.
Hint: do k binary searches (the way you binary search should be slightly different than the usual), but start the next search just after the place where the previous search terminated. Show that this can be expressed as $\sum_i \log \alpha_i$ time, where $\sum_i \alpha_i = n$ (and all α are nonnegative), then think when such a sum of logarithms is maximized.

(c) Now assume that we store the sequence x-fast tree. Show that we can process any batched predecessor query in $\mathcal{O}(k(1 + \log \log \frac{u}{k}))$ time.
Hint: The idea is really the same as in part (b): start the next search where the previous search terminated, but it is slightly more difficult to argue about what is really happening. If you don't see a formal proof, give an intuitive explanation why such complexity is the answer (or at least why we cannot hope to do better with an x-fast tree).

(SPOJ) 3. Given $n \leq 30000$ points in a plane, find the smallest distance between two of them. See "Closest pair problem" on SPOJ.

(SPOJ) 4. Given $n \leq 10^5$ points in a plane, for every point find the distance to its nearest neighbour. See "In case of failure" on SPOJ. This is a bonus question!