**Algorithms and Data Structures**
**Winter Term 2015/16**
**Exercises for Units 12 and 13**

1. If you feel uneasy with treaps, draw a few examples and test insertions and deletions.

   You can also search the web for treap/randomized search tree animations and experiment with them.

2. Let $x_1 < x_2 < \cdots < x_n$ be a sequence of $n$ keys. In a randomized search tree storing this sequence what is the expected size of the subtree whose root stores $x_k$ ?

3. Let $T$ be some binary search tree with root $r$. For node $v$ in $T$ define its *left ancestor* to be the first node on the path from $v$ to $r$ whose key is smaller than the key of $v$. Analogously the *right ancestor* is defined to be the first node on this path with larger key. Such a left ancestor or right ancestor need not exist. (Can you characterize those nodes?)

   Let us call an implementation of a binary search tree *lush* if it stores with every node four pointers: left child, right child, left ancestor, and right ancestor.

   Can rotations be realized in constant time in a lush implementation?

4. Let $\{(x_i, p_i) | 1 \leq i \leq n\}$ be a set of $n$ items with key $x_i$ and priority $p_i$ and let $x_1 < x_2 < \cdots < x_n$ and let these items be stored in a treap $T$.

   In a *finger search* for $x$ starting at item $i$ you try to find the node in $T$ with key $x$, but the search does not start at the root but at the node with key rank $i$, i.e. at the node for item $(x_i, p_i)$ (to which you assume to have pointer or "finger").

   a) Formulate such a finger search in pseudocode.

      Why would a lush implementation of the treap be useful? What would you have to change if just single parent pointers were available. What kind of sentinels would you use for the left and right parent pointers in a lush implementation?

   Now assume the $T$ is actually a randomized search tree, i.e. the priorities are i.i.d. random variables. Moreover assume that the search key $x$ happens to be $x_j$, and let $d = |j - i| + 1$. Let $D_{i,j}$ be the length of the path in $T$ between the nodes $N_i$ and $N_j$ for items $i$ and $j$.

   In a reasonable implementation of finger search the time taken is proportional to at most $D_{i,j}$. Thus we are interested in the expectation of $D_{i,j}$. In this problem you are to show that this expectation is $O(\log d)$.

   b) The lowest common ancestor $\mathrm{LCA}(a, b)$ of two nodes $a$ and $b$ in a tree is the node furthest from the root that is on the intersection of the rootpath of $a$ and the rootpath of $b$.

      Give a characterization of the lowest common ancestor $\mathrm{LCA}(N_i, N_j)$ of $N_i$ and $N_j$ in terms the sequence of priorities.

      Give a characterization of the nodes in $T$ that lie on the path from $N_i$ to $\mathrm{LCA}(N_i, N_j)$.

      Give a characterization of the nodes in $T$ that lie on the path from $N_j$ to $\mathrm{LCA}(N_i, N_j)$.

   c) Use these characterizations to show that the expected length of $D_{i,j}$ is $O(\log d)$.

5. Prove that for every treap $T$ with $n$ items there must exist a sequence of deletions so that the resulting treap has height at least $\Omega(\sqrt{n})$.

6. In the following let $p$ be some prime number, let $Z_p = \{0, 1, \ldots, p-1\}$, and let $0 < d < p$ be an integer.

   Consider the probability space $W_d = Z_p^d$ endowed with the uniform probability measure.

   For $0 \leq i < p$ let $X_i : W_d \longrightarrow Z_p$ be a the random variable defined by

$$X_i((a_0, a_1, \ldots, a_{d-1})) = a_0 + a_1 \cdot i + a_2 \cdot i^2 + \cdots + a_{d-1} \cdot i^{d-1} \bmod p\,.$$

   a) Show that for each $i$ the random variable $X_i$ has a uniform distribution, i.e. for each $r \in Z_p$ we have $\Pr(X_i = r) = 1/p$.

   b) Show that the random variables $X_0, \ldots, X_{p-1}$ are $d$-wise independent, i.e. for each $d$ distinct $i_1, i_2, \ldots, i_d$ the $d$ random variables $X_{i_1}, \ldots, X_{i_d}$ are independent.

   c) Show that if a set of random variables is $d$-wise independent, then it is also $c$-wise independet for each $c < d$.