Advanced Data Structures (spring 2007)

Exercise 4 (Wed 18.4, 12-14, C221)

Here the van Emde Boas (vEB) tree refers to the one described at the course Wiki page.

1. vEB tree — running time
   a) The running time of each operation on vEB tree can be expressed as \( T(u) = T(\sqrt{u}) + O(1) \). Show that this gives \( T(u) = O(\log \log u) \).
   b) Without the improvement of storing minimum elements as such at the bottom substructures of vEB trees, the running time of insertion and deletion can be expressed as \( T(u) = 2 \cdot T(\sqrt{u}) + O(1) \). Show that this gives \( T(u) = O(\log u) \).

2. vEB tree — space usage
   a) Why is the size of vEB tree (before using the space reduction technique) \( O(u \log \log u) \)?
   b) Consider the vEB tree with unnecessary subtrees deleted (those grayed in the Wiki page example). Can its size be expressed as a function of \( n \) so that \( u \) is a sublinear term?

3. vEB tree — pseudo code
   Write pseudo code for predecessor-query on vEB tree.

4. vEB trees and dynamic range minimum queries
   Let \( S \subseteq U \), where \( U = \{0, 1, \ldots, u - 1\} \). Each \( s \in S \) is labeled with a real value \( \ell(s) \in \mathbb{R} \). The task is to maintain a data structure on \( S \) that supports the following operations:
   - \( \text{initialize}(S) \): Construct and initialize the data structure for set \( S \).
   - \( \text{decreasekey}(s, k) \): If \( k < \ell(s) \), update the label of \( s \) to \( \ell(s) = k \). Otherwise do nothing.
   - \( \text{minimum}(r) \): return \( \min\{\ell(s) \mid s \in S, s \leq r\} \).
   Show how vEB tree can be used to supports these operations. What time complexities you obtain?