Chapter 12: Advanced Data Structures and Implementation

12.3 Incorporate an additional field for each node that indicates the size of its subtree. These fields are easy to update during a splay. This is difficult to do in a skip list.

12.6 If there are B black nodes on the path from the root to all leaves, it is easy to show by induction that there are at most 2^B leaves. Consequently, the number of black nodes on a path is at most $\log N$. Since there can't be two consecutive red nodes, the height is bounded by $2\log N$.

12.7 Color nonroot nodes red if their height is even and their parents height is odd, and black otherwise. Not all red black trees are AVL trees (since the deepest red black tree is deeper than the deepest AVL tree).

12.19 See H. N. Gabow, J. L. Bentley, and R. E. Tarjan, "Scaling and Related Techniques for Computational Geometry," *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing* (1984), 135-143, or C. Levcopoulos and O. Petersson, "Heapsort Adapted for Presorted Files," *Journal of Algorithms* 14 (1993), 395-413.

12.29 Pointers are unnecessary; we can store everything in an array. This is discussed in reference [12]. The bounds become $O(k \log N)$ for insertion, $O(k^2 \log N)$ for deletion of a minimum, $O(k^2 N)$ for creation (an improvement over the bound in [12]).

12.35 Consider the pairing heap with 1 as the root and children 2, 3, ... N. A *DeleteMin* removes 1, and the resulting pairing heap is 2 as the root with children 3, 4, ... N; the cost of this operation is N units. A subsequent *DeleteMin* sequence of 2, 3, 4, ... will take total time $\Omega(N^2)$.