Dynamic Dictionary with Working-Set Property

A binary search tree has the worst-case working-set property if every access $x_i$ costs $O(\log t_i)$ worst-case time, where $t_i$ is the number of distinct keys accessed since the last access to key $x_i$.

Describe and analyze a dynamic dictionary (not necessarily a BST) that has the working-set property. Your data structure should:

1. use $O(n)$ space, where $n$ is the current number of items in the dictionary;
2. support searching for key $x_i$ in $O(\log t_i)$ worst-case time, where $t_i$ is the number of distinct keys accessed since the insertion or last access to the key $x_i$; and
3. support insertions and deletions in $O(\log n)$ amortized time.

Hint: Consider representing your dictionary as a list of binary search trees of increasing size.