Algorithm D (*Deletion from AVL tree*). Given a set of nodes which form an AVL-balanced binary tree, this algorithm searches for a given argument K and deletes it from the tree, and rebalances the tree if necessary.

The nodes of the tree are assumed to contain KEY, LLINK, RLINK, and B fields. KEY is arbitrary data stored in a node, which must have $<$, $=$, and $>$ operations defined for it. LLINK and RLINK are pointers to the node's left and right subtrees, respectively, and either or both may be Λ. B is the height of the node's right subtree minus the height of its left subtree, and must be -1, 0, or $+1$.

The tree has a special header node in location HEAD. LLINK(HEAD) is a pointer to the root of the tree.

For convenience in description, the algorithm uses the notation $\mathtt{LINK}(a,\mathtt{P})$ as a synonym for LLINK(P) if $a = -1$, and for RLINK(P) if $a = +1$.

The algorithm makes use of a pair of auxiliary arrays named P and a, which are used as stacks. P_k is used to store pointers to nodes which may need to be rebalanced. a_k stores -1 or $+1$ such that $\mathtt{LINK}(a_k, P_k) = P_{k+1}$.

D1. [Initialize.] Set $a_0 \leftarrow -1$, $P_0 \leftarrow \mathtt{HEAD}$, $k \leftarrow 1$, $\mathtt{P} \leftarrow \mathtt{LLINK(HEAD)}$. (The pointer variable P will move down the tree.)

D2. [Compare.] If $\mathtt{K} < \mathtt{KEY(P)}$, go to D3; if $\mathtt{K} > \mathtt{KEY(P)}$, go to D4; and if $\mathtt{K} = \mathtt{KEY(P)}$, go to D5.

D3. [Move left.] Set $P_k \leftarrow \mathtt{P}$, $a_k \leftarrow -1$, $k \leftarrow k+1$, $\mathtt{P} \leftarrow \mathtt{LLINK(P)}$. If $\mathtt{P} \neq \Lambda$, go to D2; otherwise, terminate (the tree does not contain K).

D4. [Move right.] Set $P_k \leftarrow \mathtt{P}$, $a_k \leftarrow +1$, $k \leftarrow k+1$, $\mathtt{P} \leftarrow \mathtt{RLINK(P)}$. If $\mathtt{P} \neq \Lambda$, go to D2; otherwise, terminate (the tree does not contain K).

D5. [Is RLINK null?] Set $\mathtt{Q} \leftarrow \mathtt{LOC(LINK(}a_{k-1}, P_{k-1}\mathtt{))}$. (Q now points to the link that was followed to arrive at P.) If $\mathtt{RLINK(P)} \neq \Lambda$, go to D6. Otherwise, set $\mathtt{CONTENTS(Q)} \leftarrow \mathtt{LLINK(P)}$. Then, if $\mathtt{Q} \neq \Lambda$, set $\mathtt{B(CONTENTS(Q))} \leftarrow 0$, and go to D10.

D6. [Find successor.] Set $\mathtt{R} \leftarrow \mathtt{RLINK(P)}$. If $\mathtt{LLINK(R)} \neq \Lambda$, go to D7. Otherwise, set $\mathtt{LLINK(R)} \leftarrow \mathtt{LLINK(P)}$, $\mathtt{CONTENTS(Q)} \leftarrow \mathtt{R}$, $\mathtt{B(R)} \leftarrow \mathtt{B(P)}$, $a_k \leftarrow +1$, $\mathtt{P}_k \leftarrow \mathtt{R}$, $k \leftarrow k+1$, and go to D10.

D7. [Set up to find null LLINK.] Set $\mathtt{S} \leftarrow \mathtt{LLINK(R)}$, $l \leftarrow k$, $k \leftarrow k+1$, $a_k \leftarrow -1$, $P_k \leftarrow \mathtt{R}$, $k \leftarrow k+1$.

D8. [Find null LLINK.] If $\mathtt{LLINK(S)} = \Lambda$, go to D9. Otherwise, set $\mathtt{R} \leftarrow \mathtt{S}$, $\mathtt{S} \leftarrow \mathtt{LLINK(R)}$, $a_k \leftarrow -1$, $P_k \leftarrow \mathtt{R}$, $k \leftarrow k+1$, and repeat this step.

D9. [Fix up.] Set $a_l \leftarrow +1$, $P_l \leftarrow \mathtt{S}$, $\mathtt{LLINK(S)} \leftarrow \mathtt{LLINK(P)}$, $\mathtt{LLINK(R)} \leftarrow \mathtt{RLINK(S)}$, $\mathtt{RLINK(S)} \leftarrow \mathtt{RLINK(P)}$, $\mathtt{B(S)} \leftarrow \mathtt{B(P)}$, and $\mathtt{CONTENTS(Q)} \leftarrow \mathtt{S}$.

D10. [Adjust balance factors.] Set $k \leftarrow k-1$. If $k = 0$ then terminate successfully. Otherwise, set $\mathtt{S} \leftarrow P_k$, and consider the following cases:

i) If $\mathtt{B(S)} = 0$, set $\mathtt{B(S)} \leftarrow -a_k$, and terminate successfully.

ii) If $\mathtt{B(S)} = a_k$, set $\mathtt{B(S)} \leftarrow 0$, and repeat this step.

iii) Otherwise, $\mathtt{B(S)} = -a_k$. Set $\mathtt{R} \leftarrow \mathtt{LINK(}-a_k\mathtt{,S)}$. Go to D11 if $\mathtt{B(R)} = 0$; go to D12 if $\mathtt{B(R)} = -a_k$; otherwise, $\mathtt{B(R)} = a_k$, and go to D13.

D11. [Single rotation with balanced R.] Set $\mathtt{LINK(}-a_k\mathtt{,S)} \leftarrow \mathtt{LINK(}a_k\mathtt{,R)}$, $\mathtt{LINK(}a_k\mathtt{,R)} \leftarrow \mathtt{S}$, $\mathtt{B(R)} \leftarrow a_k$, $\mathtt{LINK(}a_{k-1}, P_{k-1}\mathtt{)} \leftarrow \mathtt{R}$, and terminate successfully.

D12. [Single rotation with unbalanced R.] Set $\mathtt{LINK(}-a_k\mathtt{,S)} \leftarrow \mathtt{LINK(}a_k\mathtt{,R)}$, $\mathtt{LINK(}a_k\mathtt{,R)} \leftarrow \mathtt{S}$, $\mathtt{B(S)} \leftarrow \mathtt{B(R)} \leftarrow 0$, $\mathtt{LINK(}a_{k-1}\mathtt{,P}_{k-1}\mathtt{)} \leftarrow \mathtt{R}$, and go to D10.

D13. [Double rotation.] Set $\mathtt{P} \leftarrow \mathtt{LINK(}a_k\mathtt{,\ R)}$, $\mathtt{LINK(}a_k\mathtt{,\ R)} \leftarrow \mathtt{LINK(}-a_k\mathtt{,\ P)}$, $\mathtt{LINK(}-a_k\mathtt{,\ P)} \leftarrow \mathtt{R}$, $\mathtt{LINK(}-a_k\mathtt{,S)} \leftarrow \mathtt{LINK(}a_k\mathtt{,P)}$, $\mathtt{LINK(}a_k\mathtt{,P)} \leftarrow \mathtt{S}$. Update balance factors as follows:

i) If $\mathtt{B(P)} = -a_k$, set $\mathtt{B(S)} \leftarrow a_k$ and $\mathtt{B(R)} \leftarrow 0$.

ii) If $\mathtt{B(P)} = 0$, set $\mathtt{B(S)} \leftarrow \mathtt{B(R)} \leftarrow 0$.

iii) Otherwise, $\mathtt{B(P)} = a_k$. Set $\mathtt{B(S)} \leftarrow 0$ and $\mathtt{B(R)} \leftarrow -a_k$.

Finally, set $\mathtt{B(P)} \leftarrow 0$ and $\mathtt{LINK(}a_{k-1}\mathtt{,P}_{k-1}\mathtt{)} \leftarrow \mathtt{P}$, and go to D10. ∎