
```

procedure Insert( $x$ );
     $count \leftarrow count + 1$ ;
    if  $count > M$ 
    then
        RehashAll( $x$ );
    else
         $j \leftarrow h(x)$ ;
        if position  $h_j(x)$  of subtable  $T_j$  contains  $x$ 
        then
            if  $x$  is marked “deleted” then remove this tag;
        else (*  $x$  is new for  $W_j$  *)
             $b_j \leftarrow b_j + 1$ ;
            if  $b_j \leq m_j$ 
            then (* size of  $T_j$  sufficient *)
                if position  $h_j(x)$  of  $T_j$  is empty
                then
                    store  $x$  in position  $h_j(x)$  of  $T_j$ ;
                else
                    go through the subtable  $T_j$ , put all elements
                    not marked “deleted” into a list  $L_j$ , and
                    mark all positions of  $T_j$  empty;
                    append  $x$  to list  $L_j$ ;  $b_j \leftarrow \text{length of } L_j$ ;
                    repeat  $h_j \leftarrow$  randomly chosen function in  $\mathcal{H}_{s_j}$ 
                    until  $h_j$  is injective on the elements of list  $L_j$ ;
                    for all  $y$  on list  $L_j$  store  $y$  in position  $h_j(y)$  of  $T_j$ ;
            else (*  $T_j$  is too small *)
                 $m_j \leftarrow 2 \cdot \max\{1, m_j\}$ ;  $s_j \leftarrow 2m_j(m_j - 1)$ ;
                if condition (**) is still satisfied
                then (* double capacity of  $T_j$  *)
                    allocate new space, namely  $s_j$  cells, for new subtable  $T_j$ ;
                    go through old subtable  $T_j$ , put all elements
                    not marked “deleted” into a list  $L_j$ ,
                    and mark all positions empty;
                    append  $x$  to list  $L_j$ ;  $b_j \leftarrow \text{length of } L_j$ ;
                    repeat  $h_j \leftarrow$  randomly chosen function in  $\mathcal{H}_{s_j}$ 
                    until  $h_j$  is injective on the elements of list  $L_j$ ;
                    for all  $y$  on list  $L_j$  store  $y$  in position  $h_j(y)$  of  $T_j$ ;
            else (* level-1-function  $h$  “bad” *)
                RehashAll( $x$ );

```

Figure 1: Insertion

```

procedure RehashAll( $x$ );
    (* RehashAll( $x$ ) is either called by Insert( $x$ ), and then  $x \in U$ ,
       or by Delete( $x$ ), and then  $x = -1$ . RehashAll( $x$ ) builds a new table
       for all elements currently in the table plus  $x$  (if  $x \in U$ ). *)
    go through the whole table  $T$ , put all elements not tagged "deleted"
    into a list  $L$ , count them, and mark all positions in  $T$  "empty";
    if  $x \in U$  then append  $x$  to  $L$ ;
     $count \leftarrow$  length of list  $L$ ;
     $M \leftarrow (1 + c) \cdot \max\{count, 4\}$ ;
    repeat  $h \leftarrow$  randomly chosen function in  $\mathcal{H}_{s(M)}$ ;
        for all  $j$ ,  $0 \leq j < s(M)$ , do form a list  $L_j$  of all  $x \in L$  with  $h(x) = j$ ;
        for all  $j$ ,  $0 \leq j < s(M)$ , do
             $b_j \leftarrow$  length of list  $L_j$ ;  $m_j \leftarrow 2 \cdot b_j$ ;  $s_j \leftarrow 2m_j(m_j - 1)$ ;
    until condition (**) is satisfied;
    for all  $j$ ,  $0 \leq j < s(M)$ , do
        allocate space  $s_j$  for subtable  $T_j$ ;
        repeat  $h_j \leftarrow$  randomly chosen function in  $\mathcal{H}_{s_j}$ ;
            until  $h_j$  is injective on the elements of list  $L_j$ ;
        for all  $x$  on list  $L_j$  do store  $x$  in position  $h_j(x)$  of  $T_j$ ;

procedure Delete( $x$ );
     $count \leftarrow count + 1$ ;
     $j \leftarrow h(x)$ ;
    if position  $h_j(x)$  of subtable  $T_j$  contains  $x$ 
        then mark  $x$  as "deleted"
        else return( $x$  is not a member of  $S$ );
    if  $count \geq M$ 
        then (* start new phase *)
            RehashAll( $-1$ );

procedure Lookup( $x$ );
     $j \leftarrow h(x)$ ;
    if position  $h_j(x)$  of subtable  $T_j$  contains  $x$  (not marked "deleted")
        then return(" $x$  is a member of  $S$ ")
        else return(" $x$  is not a member of  $S$ ");

procedure Initialize;
     $T \leftarrow$  an empty table;
    RehashAll;

```

Figure 2: Setup, Deletion, Lookup, and Rehashing