S. Schwoon / D. Suwimonteerabuth Winter Term 2008/09

Fundamental Algorithms

Solution Keys 6

1. See exercise sheet 5.

2. (a) **Procedure** `findMin`

Input: a pointer p to a binary search tree
Output: a pointer to a node containing the smallest value

if $p =$ `NIL` **then return** `NIL`;
while $p \rightarrow$ `.left` $\neq$ `NIL` **do**
    $p := p \rightarrow$ `.left`;
od
return p;

(b) **Procedure** `remove`

Input: a pointer p to a binary search tree, an integer x
Output: a pointer to the tree, from which a node with value x is removed

if $p =$ `NIL` **then return** `NIL`;
if $p \rightarrow$ `.value` $< x$ **then**
    $p \rightarrow$ `.right` $:=$ `remove`$(p \rightarrow$ `.right`$, x)$;
else if $p \rightarrow$ `.value` $> x$ **then**
    $p \rightarrow$ `.left` $:=$ `remove`$(p \rightarrow$ `.left`$, x)$;
else if $p \rightarrow$ `.left` $\neq$ `NIL` **and** $p \rightarrow$ `.right` $\neq$ `NIL` **then**
    $p \rightarrow$ `.value` $:=$ `findMin`$(p \rightarrow$ `.right`$) \rightarrow$ `.value`;
    $p \rightarrow$ `.right` $:=$ `remove`$(p \rightarrow$ `.right`$, p \rightarrow$ `.value`$)$;
else if $p \rightarrow$ `.left` $\neq$ `NIL` **then**
    $p := p \rightarrow$ `.left`;
else
    $p := p \rightarrow$ `.right`;
fi
return p;

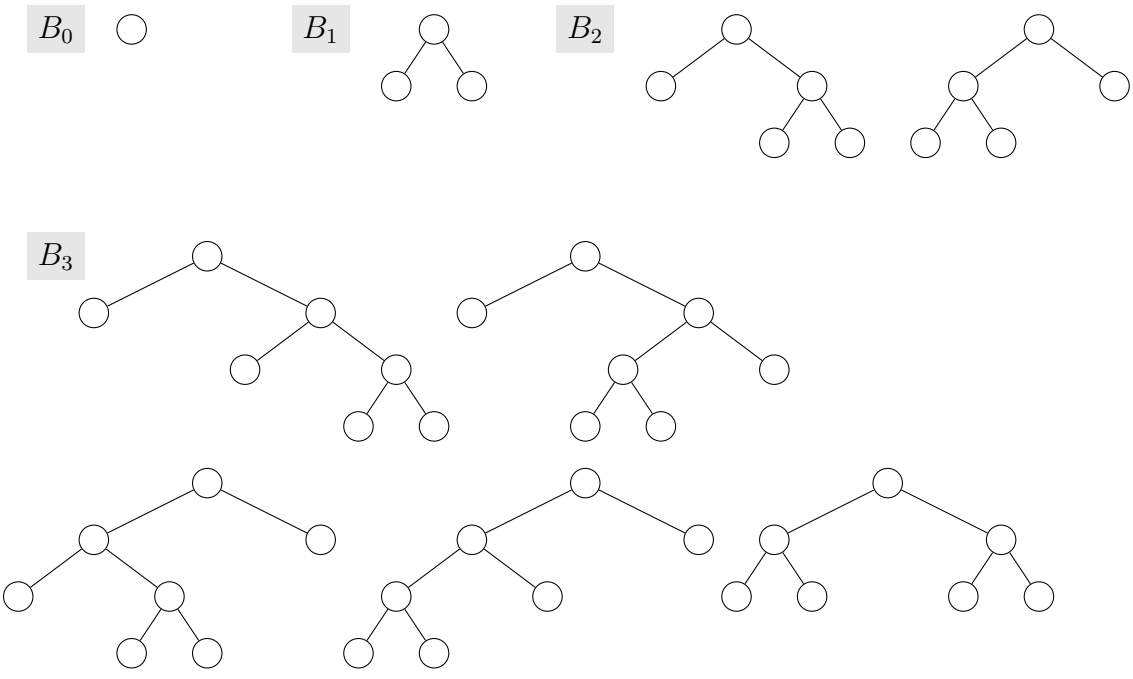
3. (a) h and $2^h - 1$, respectively.

(b) **Procedure** `binary_tree_insert`

Input: a pointer p to a binary search tree, an integer k
Output: a pointer to tree, to which a node with value k is inserted

if $p =$ `NIL` **then**
  $p :=$ `newBinNode`;
  $p \rightarrow$ `.value` $:= k$; $p \rightarrow$ `.height` $:= 1$;
  $p \rightarrow$ `.left` $:=$ `NIL`; $p \rightarrow$ `.right` $:=$ `NIL`;
  return p;
fi
if $p \rightarrow$ `.value` $> k$ **then**
  $p \rightarrow$ `.left` $:=$ `binary_tree_insert`$(p \rightarrow$ `.left`$, k)$;
else
  $p \rightarrow$ `.right` $:=$ `binary_tree_insert`$(p \rightarrow$ `.right`$, k)$;
fi
if $p \rightarrow$ `.left` $=$ `NIL` **then**
  $p \rightarrow$ `.height` $:= 1 + p \rightarrow$ `.right` $\rightarrow$ `.height`;
else if $p \rightarrow$ `.right` $=$ `NIL` **then**
  $p \rightarrow$ `.height` $:= 1 + p \rightarrow$ `.left` $\rightarrow$ `.height`;
else
  $p \rightarrow$ `.height` $:= 1 + \max(p \rightarrow$ `.left` $\rightarrow$ `.height`$, p \rightarrow$ `.right` $\rightarrow$ `.height`$)$;
fi
return p;

4. (a)



(b) $|B_n| = \sum_{i=0}^{n-1} |B_i||B_{n-1-i}|$, where $|B_0| = 1$.