Binäre Bäume

8. Aufgabe *10 Punkte*

Implementieren Sie die Schnittstellen für folgenden binären Baum.

```
typedef struct BiTreeNode_Struct {

int                         count;
struct BiTreeNode_Struct    *left;
struct BiTreeNode_Struct    *right;
} BiTreeNode;

typedef struct BiTree_Struct {

int                         size;
BiTreeNode                  *root;

} BiTree;
```

void **bitree_init**(BiTree *tree);

void **bitree_destroy**(BiTree *tree);

int **bitree_insert_left**(BiTree *tree, BiTreeNode *node, int count);

int **bitree_insert_right**(BiTree *tree, BiTreeNode *node, int count);

void **bitree_remove_left**(BiTree *tree, BiTreeNode *node);

void **bitree_remove_right**(BiTree *tree, BiTreeNode *node);

Zu den Funktionen *bitree_insert_right / bitree_insert_left*:
Falls der Knoten *node* NULL ist und die Baumgröße gleich Null ist, wird an der Wurzel eingefügt. Ansonsten wird der neue Knoten als rechter / linker Nachfolger des Knoten *node* eingehängt.

Zu den Funktionen *bitree_remove_right / bitree_remove_left*:
Der rechte bzw. der linke Unterbaum des Knoten *node* wird gelöscht. Falls der Knoten *node* NULL ist, wird mit dem Löschen an der Wurzel begonnen.

Ein Beispielprogramm, mit welchem Sie Ihre Schnittstellen testen können, finden Sie unter *http://www.mathematik.uni-ulm.de/sai/ws99/soft/blatt4/bitree-ex-1.c*.

Susanne Schmucker, 23.11.1999