DSA – Practical
Trees Operations 2

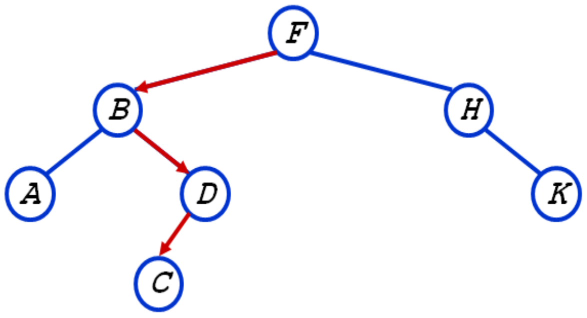
Continue what you worked on in previous week to implement other trees operations:

1. Write the implementation of the following functions:

```
// deletes the node with x value in the tree rooted at root, and returns
the root of the modified tree after deletion, as root itself might be
deleted and thus the root changed
BinaryTreeNode* bstDelete(BinaryTreeNode* root, double x) {
       …
}

// Returns 1 or 0 depending on the given tree being AVL or not
int bstIsAVL(BinaryTreeNode* root) {
       …
}
```

- Note that deleting a node (target node) must consider three cases:
  - Case 1 : The target node has no children (K or C)
    - Remove and free the target (through parent)
  - Case 2 : The target node has one child (H or D)
    - Make target's parent point to target's child and free target
  - Case 3 : The target node has two children (B or F)
    - Find smallest node (minimum value) in target's right subtree, and put it in place of the target, then free the target



2. Test the above functions with something like the following main() function:

```
void main(){
	BinaryTreeNode* node;
	BinaryTreeNode* root = (BinaryTreeNode*) malloc(sizeof(BinaryTreeNode));
	root->key = 5.5;
	root->leftChild = NULL;
	root->rightChild = NULL;

	bstInsert(root, 7.7);
	bstInsert(root, 3.2);
	bstInsert(root, 4.0);
	...

	node = bstFind(root, 3.2);
	if(node != NULL)
		printf("The value %f was found.\n", node->key);
	else
		printf("Value 3.2 was not found.\n";

	node = bstMin(root);
	if(node != NULL)
		printf("The minimum value is %f.\n", node->key);

	node = bstMax(root);
	if(node != NULL)
		printf("The maximum value is %f.\n", node->key);

	root = bstDelete(root, 3.2);
	node = bstFind(root, 3.2);
	if(node != NULL)
		printf("The value %f was found.\n", node->key);
	else
		printf("Value 3.2 was not found.\n";

	// delete the root
	root = bstDelete(root, 7.7);

	printf("Height of binary search tree is %d.\n", bstHeight(root));

	if(bstIsAVL(root))
		printf("The tree is an AVL tree.\n");
	else
		printf("The tree is NOT an AVL tree.\n";

	bstClear(root);
	root = NULL;
}
```