1) (10 pts) DSN (Binary Trees)

A binary search tree is considered "lopsided" if the root's left subtree height and right subtree height differ by more than one (i.e., the left subtree is more than one level deeper or shallower than the right subtree). This is different from the definition of "balanced" that comes up in relation to AVL trees, because the "lopsided" property only applies to the root of the tree – not every single node in the tree.

Write a function, *isLopsided()*, that takes the root of a binary search tree and returns 1 if the tree is lopsided, and 0 otherwise. You may write helper functions as you see fit. The node struct and function signature are as follows:

```
typedef struct node {
   struct node *left, *right;
   int data;
} node;

int isLopsided(node *root) {




}
```