Recursion on Tree Practice

21. Write a function f21(tree) that returns the height of the tree. The tree has the structure [value, left subtree, right subtree].

```
>>> f21([])
0

>>> f21([1,[],[]])
1

>>> f21([1, [1,[],[]], []])
2
```

22. Write a function f22(tree) that returns the number of nodes in the tree. The tree has the structure [value, left subtree, right subtree].

```
>>> f22([])
0

>>> f22([1,[],[]])
1

>>> f22([1, [1,[],[]], [1,[],[]])
3
```

23. Write a function f23(tree) that returns the sum of the nodes in the tree. The tree has the structure [value, left subtree, right subtree].

```
>>> f23([])
0

>>> f23([1,[],[]])
1

>>> f23([1, [2,[],[]], [3,[],[]] )
6
```

24. Write a function f24(tree) that prints out the values of the tree in ascending order. The tree has the structure [value, left subtree, right subtree] and is a binary search tree.

```
>>> f24([])

>>> f24([1,[],[]])
1

>>> f24( [2,[1,[],[]], [3,[],[4,[],[]] ])
1
2
3
4
```

25. Write a function f25(tree) that returns the smallest element in the tree. The tree has the structure [value, left subtree, right subtree] and is a binary search tree. Return -1 if the tree is empty.

```
>>> f25([])
-1

>>> f25([1,[],[]])
1

>>> f25([2,[1,[],[]],[3,[],[4,[],[]]]])
1
```