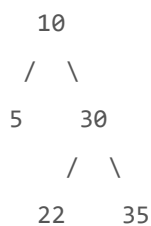# Daily Coding Problem #133

Problem

This problem was asked by Amazon.

Given a node in a binary search tree, return the next bigger element, also known as the inorder successor.
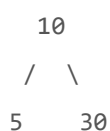
For example, the inorder successor of 22 is 30.

```
    10
   /  \
  5    30
      /  \
    22    35
```

You can assume each node has a `parent` pointer.

Solution

We can use case-analysis to break the problem down to two steps.

- First, if there is a right child of `node`, then the leftmost descendant of `node.right` (or just `node.right` if it has none) is simply the inorder successor.
- Otherwise, we can find the inorder successor by traversing through the parent pointers, keeping track of the current `node` and `parent`. When we find a `parent` whose left child is equal to `node`, then we know this is the inorder successor.

Let's look at an example.

```
    10
   /  \
  5    30
```

```
  /  \
22    35
  \
  25
```

- The inorder successor of 10 is 22 since it has a right child and 22 is the leftmost child of node.right.
- The inorder successor of 25 is 30 since 30 is the first parent where parent.left is node.

```python
class Node:
    def __init__(self, val, left=None, right=None, parent=None):
        self.val = val
        self.left = left
        self.right = right
        self.parent = parent


def inorder_successor(node):
    if node.right:
        return leftmost(node.right)

    parent = node.parent

    while parent and parent.left is not node:
        parent, node = parent.parent, parent

    return parent

def leftmost(node):
    while node.left:
        node = node.left
    return node
```