1. Write a `recursive` method `return_max` which returns the maximum value stored in a BST. Raise a `ValueError` if the BST is empty.
2. We will now look at a mutating method: deleting an item from a BST. Assume you have a method delete_root, which deletes the root of a BST. Using it, implement the method below:

```python
def delete_item(self, item):
    """Delete one occurrence of <item> in this BST.
    Do nothing if <item> is not in this BST.
    @type self: BinarySearchTree
    @type item: object
    @rtype: None
    ""
```

Now let's work on delete_root. Here is a diagram of a fairly large BST.

```
        10
       / \
      3   32
     / \ /  \  
    2  7 27 81
     \ /      /  
      49   99
```

3. Suppose we want to delete the root value (10) and replace it with another value in the tree. What are the two possible values we could replace it with and still have a valid BST?
4. Remember your work on the first question. Create a new method `extract_max` which not only returns the maximum value in a BST, but also removes it from the tree.

5. Using `extract_max` as a helper method (and possibly defining other helper methods), implement `delete_root`. 