Definitions used for the Midterm Exams 132–all sections

Common Definition(s)

Binary Trees

```java
public class BTree< T > {
    public boolean isEmpty();
    public boolean isLeaf();
    public T getValue();
    public BTree< T > getLeft();
    public BTree< T > getRight();
}
```

<table>
<thead>
<tr>
<th>Method Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isEmpty()</td>
<td>Returns true if this BTree is empty.</td>
</tr>
<tr>
<td>boolean isLeaf()</td>
<td>Returns true if this BTree has empty Binary Trees for its left and right siblings.</td>
</tr>
<tr>
<td>T getValue()</td>
<td>Returns the value on this BTree object; note, an error results if this tree is empty.</td>
</tr>
<tr>
<td>BTree&lt;T&gt; getLeft()</td>
<td>Returns the left child. Note: calling this on an empty tree throws an exception.</td>
</tr>
<tr>
<td>BTree&lt;T&gt; getRight()</td>
<td>Returns the right child. Note: calling this on an empty tree throws an exception.</td>
</tr>
</tbody>
</table>

Definitions of commonly used terms

**Definition** (Edge). An *edge* in a binary tree is a binary relation that takes a *parent* (or superordinate) node to a *child* (or subordinate) node.

**Definition** (Internal node(s)). An *internal node* in a binary tree is any node that has at least one subordinate (child) node.

**Definition** (Leaf node(s)). A *leaf node* in a binary tree is any node that has no subordinate (child) nodes.
Provided methods, classes, etc.

You should assume that each of these methods works as described when answering any questions regarding Binary Trees on this exam. Do not add any properties to these definitions.

You may assume that any of the common arithmetic operators are available as are the functions max and min which assume their usual meanings.

Prohibited usage

- No question on this exam permits the use of any of Java’s exception machinery.
- No use of iteration where recursion is required is allowed.
- No use of additional Java classes, such as those from the util package is allowed, such as any Collections classes.