Consider binary trees, represented as BinTree objects and whose nodes are represented as Node objects. Both have the expected structure: the BinTree object points to a root Node object. Each Node object has left, right and parent pointers to its left child, right child and parent nodes respectively; these can be null. Additionally, each node has a container pointer to the (unique) BinTree that contains it, and has key and value fields. There are no duplicate keys within a BinTree.

Now assume functions: nodes(T) which gives the set of all nodes in BinTree T, and descendants(n) which gives the set of nodes reachable by following zero-or-more left and right links from Node n.

Let BT1 and BT2 be properties of a BinTree T, defined by:

\[
\begin{align*}
\text{BT1}(T) & \iff \\
\forall n \in \text{nodes}(T) : n.\text{left} \neq \text{null} \Rightarrow n.\text{left}.\text{key} < n.\text{key} \\
& \land \forall n \in \text{nodes}(T) : n.\text{right} \neq \text{null} \Rightarrow n.\text{key} < n.\text{right}.\text{key}.
\end{align*}
\]

\[
\begin{align*}
\text{BT2}(T) & \iff \\
\forall n \in \text{nodes}(T), \forall m \in \text{descendants}(n.\text{left}) : m.\text{key} < n.\text{key} \\
& \land \forall n \in \text{nodes}(T), \forall q \in \text{descendants}(n.\text{right}) : n.\text{key} < q.\text{key}.
\end{align*}
\]

(a) State whether each of the following statements is true or false, justifying your statement with a proof or counterexample.

(i) \( \forall \text{BinTree } T : \text{BT1}(T) \Rightarrow \text{BT2}(T) \). [2 marks]

(ii) \( \forall \text{BinTree } T : \text{BT2}(T) \Rightarrow \text{BT1}(T) \). [2 marks]

(b) Write neat and well-commented pseudocode for a void method deleteRoot of class BinTree. When given a BinTree T that satisfies BT2(T), the method must delete the root note and rearrange T so that it continues to satisfy BT2(T). It is an explicit requirement that you delete the root node of T, as opposed to deleting some other node and copying its key and value into the former root. [Hint: sketching diagrams of the various cases and referring to them in comments is not required but may help you write correct pseudocode.]

[7 marks for correctness + 7 marks for clarity]

(c) What undesirable consequences might ensue if a programmer violated the explicit requirement specified in Part (b)? [2 marks]