1. (a) [5 marks] Prove or disprove: \(2^n \in O(2^n)\).
   (b) [5 marks] Prove or disprove: if \(f(n) \in O(g(n))\), then \(2^{f(n)} \in O(2^{g(n)})\).

2. This question is about proving a useful theorem from the textbook:
   If \(d(n) \in O(f(n))\) and \(e(n) \in O(g(n))\), then \(d(n) \cdot e(n) \in O(f(n) \cdot g(n))\).
   (a) [2 marks] My friend attempts the proof below. What goes wrong?
   Suppose \(d(n) \in O(f(n))\) and \(e(n) \in O(g(n))\).
   Then \(\lim_{n \to \infty} \frac{d(n)}{f(n)}\) and \(\lim_{n \to \infty} \frac{e(n)}{g(n)}\) exist and are finite.
   Hence
   \[
   \left(\lim_{n \to \infty} \frac{d(n)}{f(n)}\right) \left(\lim_{n \to \infty} \frac{e(n)}{g(n)}\right) = \lim_{n \to \infty} \frac{d(n) \cdot e(n)}{f(n) \cdot g(n)}
   \]
   exists and is finite.
   Therefore \(d(n) \cdot e(n) \in O(f(n) \cdot g(n))\).
   (b) [10 marks] Prove the theorem.

3. [5 marks] Build an AVL tree from the following keys. Start with empty and insert the keys (and rebalance after each insertion) in the order listed. Just hand in the final tree.
   17, 84, 76, 41, 59, 13, 11, 16, 62, 54, 43

4. [5 marks] Remove 75 from the following AVL tree. Rebalance. Hand in: (i) the tree after removing but before rebalancing; (ii) the tree after rebalancing.

5. Here are two algorithms for finding the largest key in a binary search tree. Prove them correct the CSCB36 way.
   (a) [10 marks] Iteratively (you may omit termination):
   Precondition: \(\text{root}\) is not empty (null).
   Postcondition: \(r\) is the largest key in the binary search tree at \(\text{root}\).
   ```java
   v = root;
   while (v.right != null) {
       v = v.right;
   }
   r = v.key;
   ```
   (b) [10 marks] Recursively:
   Precondition: \(v\) is not empty (null).
   Postcondition: \(\text{maxaux}(v)\) returns the largest key in the binary search tree at \(v\).
private K maxaux(Node<K> v) {
    return v.right == null ? v.key : maxaux(v.right);
}

public K max() {
    return maxaux(root);
}

6. In an AVL tree, although the left subtree and the right subtree differ in heights by at most 1, their difference in the numbers of nodes can be huge. This question investigates the maximum difference possible.

(a) [3 marks] Draw an AVL tree of height 5 that has the minimum possible number of nodes in the left subtree and the maximum possible in the right subtree. You may omit the keys.

(b) [5 marks] If an AVL tree has height \( h \) (assume \( h \geq 2 \)), what is the maximum possible difference between the number of nodes in its two subtrees? Prove your answer. Your answer should not use big-Oh or big-Theta.

(c) [10 marks] If an AVL tree has \( n \) nodes (assume \( n \) is large enough), what is the maximum possible ratio between the number of nodes in its two subtrees? Prove your answer. Your answer may use big-Oh or big-Theta.