1. Given a list of 25000 names, ordered alphabetically. We search the list for a certain name in a sequential fashion. How many comparisons are needed in the worst and best case? And assuming the name is present in the list, how many comparisons are needed in the average case?

2. Consider the sorted array

\[3 \ 14 \ 27 \ 31 \ 39 \ 42 \ 55 \ 70 \ 74 \ 81 \ 85 \ 93 \ 98\]

Determine for each number in this array how many comparisons are needed to find it with a binary search.

3. Sequential search can be implemented with about the same efficiency on an array or on a linked list of numbers. Does the same hold for binary search?

4. Which of the following binary trees are AVL trees?

(a) 

```
    5
   / \  
  3   6
 / \  /  
2  8 7  9
```

(b) 

```
    5
   / \  
  4   6
 / \  /  
2  8 7  9
```
5. (a) For each $n \leq 5$ draw all binary trees with $n$ nodes that satisfy the balance requirements of AVL trees.

(b) Draw an AVL tree of depth 4 that has the smallest number of nodes among all such trees.

6. (a) For each of the following sequences construct an AVL tree by successively inserting the numbers in the list, starting from the empty tree:

   * 1 2 3 4 5 6
   * 1 5 4 3 2

   After each insertion, restructure the tree into an AVL tree if needed.

(b) In both cases, remove the root from the final AVL tree, and restructure the resulting binary search tree into an AVL tree again if needed.

7. Design an algorithm to determine the smallest number in an AVL tree.
   What is the worst-case time complexity of your algorithm?

8. True or false: The smallest and largest number in an AVL tree can always be found on either the last or the one-before-last level?

9. Consider below the Left-Right case of adding a node to an AVL tree. The root of the depicted subtree is the lowest unbalanced node, with balance factor $-2$, because the depth of subtree has $B$ increased, say from $h$ to $h + 1$.

   ![Diagram](image)

   Argue that subtrees $A$ and $C$ both have depth $h$. 