1. (a) Insert the following sequence of elements in an AVL tree, starting with an empty tree. Show each rebalancing that needs to be done and the result after rebalancing.

\[(10, 20, 50, 30, 60, 40, 5, 90, 55, 95)\]

(b) Perform a **postorder traversal** of the final AVL tree and print the result. What is the time complexity of such a traversal for an AVL tree of \(n\) nodes?

(c) Suppose a postorder traversal of a given AVL tree produces the following sequence. Construct the AVL tree from this postorder list. Briefly explain how this is done.

\[(10, 25, 40, 30, 20, 60, 80, 70, 50)\]
2. Consider sorting $n = 6$ real numbers.

   (a) Derive a lower bound (exact value) for the worst-case number of key comparisons needed.

   (b) Suppose we apply insertion sort for this problem. Derive the worst-case number of key comparisons (exact value).

   (c) Draw a diagram to show how mergesort may be applied to this problem. Derive the worst-case number of key comparisons (exact value).
3. Given an array $A$ of $n$ real numbers. Suppose the array has many repetitions, so that the array consists of only $k$ distinct values, for some $k \leq \log n$.

Describe in simple words an algorithm to find all distinct values in sorted order. The final sorted result should be placed in a second array $B$. Analyze the time complexity in terms of $n$ and $k$. 
4. Given an undirected graph with $n$ vertices and $e$ edges. The graph is represented by its adjacency lists. Describe an algorithm to find the connected components of the graph. The result must be in an array $C$ such that for every vertex $i$ in component number $k$, $C[i] = k$. (You may use the index of the smallest vertex in each component to number that component.)

Analyze the time complexity of the algorithm in terms of $n$ and $e$. 
5. Floyd’s all-pairs-shortest-paths algorithm works on a directed weighted graph and finds the least-cost (shortest) path between every pair of vertices \((i, j)\). The algorithm produces two matrices \(A\) (cost) and \(P\) (path), such that at the end, \(A[i, j]\) is the cost of the shortest \((i, j)\) path, and \(P[i, j]\) is the first vertex immediately after vertex \(i\) on the shortest-path.

(a) What is the time complexity of this algorithm? (State the appropriate graph representation.)

(b) Show the working of Floyd’s algorithm for the following weighted graph. Show the matrix \(A\) and \(P\) after each iteration.