1. Consider the following recurrence relation, where $n$ is a power of 2:

$$T(n) = \begin{cases} 
1, & n = 2 \\
2T(n/2) + n, & n > 2 
\end{cases}$$

Prove by **induction** that the solution is $T(n) \leq An \log n$, and determine the constant $A$.
2. (a) Starting with an empty AVL tree, perform the following sequence of INSERT operations:

20, 10, 4, 5, 3, 7

Show the result immediately after each insert operation, mark any rebalancing needed, and then the result after rebalancing.

(b) In the following AVL tree, delete element 60. (First complete the picture by drawing lines from each node to its children.) If any rebalancing is needed, mark the nearest unbalanced ancestor and the type of rebalancing to be done, and the result after that rebalancing operation.

```
     50
   /   \
  40    70
 /     /  \
30    42  60  80
 /  \
20  41  46  90
  \
    44  48
```
3. (a) State the lower bound we proved for comparison-based sorting algorithms. (State the result without proof.)

(b) What algorithm would you use to sort \( n \) integers, each in the range 0 to \( n - 1 \)? (Name the algorithm and state its time complexity.)

Briefly explain why the above general lower bound does not apply to this case.

(c) How would you sort \( n \) integers, each in the range 0 to \( n^3 - 1 \)? Outline the sorting algorithm and analyze its time complexity.
4. (a) Consider the problem of finding median of \( n \) elements. For each of the following cases, state an efficient algorithm and its time complexity. (For each case, we want the simplest algorithm that achieves the most-efficient worst-case time complexity.)

i. Assume \( n \) integers, each in the range 0 to 100.

ii. Assume \( n \) real-valued elements.

(b) Consider a sorted array \( A[0..n-1] \) of \( n \) elements. We want to determine a value \( C \) that minimizes the following sum \( S \) (where \( | \cdot | \) denotes the absolute-value).

\[
S = \sum_{i=0}^{n-1} |(A[i] - C)|,
\]

i. Briefly prove that \( S \) is minimized when \( C = \) median element. (Assume \( n \) is odd.)

ii. Compute the minimum sum \( S \) for the following sequence of elements. (Show the \( C \) value and the computation of sum \( S \)).

\[
10, 20, 50, 60, 100
\]
5. Find a minimum-cost-spanning tree (MST) of the following weighted undirected graph. Use PRIM’s algorithm, starting with vertex 1 as the root. Show the result after each iteration, and the final MST and its cost. (Assume the graph is represented by its adjacency lists, and the vertices in each list are in increasing order of vertex number.)

![Graph Diagram]

What is the time complexity of PRIM’s algorithm (when the graph is represented by its adjacency lists)? Briefly explain the analysis.