1. Consider the following recurrence.

\[ T(n) \leq \begin{cases} 
T(3n/4) + T(n/8) + 4n, & n > 16 \\
n^2, & n \leq 16 
\end{cases} \]

Prove by induction that the solution is as follows, and determine the constant \( A \).

\[ T(n) \leq An, \quad \text{for all } n \geq 1 \]

<table>
<thead>
<tr>
<th>GRADE</th>
<th>/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>SUM</td>
<td>/100</td>
</tr>
</tbody>
</table>
2. (a) Starting with an empty AVL tree, perform the following sequence of insertions. Show the resulting AVL tree after each insertion.

10, 5, 20, 15, 30, 40, 50, 25, 22, 26, 28.

(b) Delete element 40 in the following AVL tree.

```
    50
   /   \
  40    70
 /  \
30   60  80
```

(c) What is the worst-case time complexity of each of the following operations in an AVL tree.

- Search
- Insert
- Delete
- Delete the element with the largest key. (Provide a brief explanation.)
3. (a) Describe an efficient algorithm for sorting $n$ integers, where each integer is represented as a $(3 \log n)$-bit binary integer. Analyze the time complexity.

(b) Consider a directed, weighted graph with $n$ vertices. Suppose the outdegree of each vertex is $d$, for some constant $d$. What graph representation is appropriate for Dijkstra's single-source shortest-path algorithm, and what is the resulting time complexity?

(c) Suppose we want to find a depth-first spanning forest for the graph of part (b). What is the running time of the algorithm?
4. (a) Show the working of Dijkstra’s single-source shortest-path algorithm for the following directed, weighted graph. Use vertex 1 as the source.

(b) Show the working of Floyd’s all-pairs-shortest-path algorithm on the above graph. Show the adjacency matrix at the beginning and the resulting matrix at the end of each iteration.

(c) What is the worst-case time complexity of each of the following graph algorithms, assuming the graph is represented by its adjacency matrix.
   i. Depth-First search
   ii. Dijkstra’s single-source shortest paths
   iii. Floyd’s all-pairs shortest-path algorithm.
5. Given an array of \( n \) elements with arbitrary values. Describe how to find the \( \sqrt{n} \) smallest elements in sorted order, using each of the following methods. For each case, outline the algorithm in terms of the major steps involved, and analyze the worst-case time complexity.

(a) Use a heap.

(b) Use the selection algorithm with the “median-of-medians” pivot, as described in class.