Q1a. What is the value of count[j] of the following pseudo-code fragment when n=2:
   1. For i = 1 to n do
   2.   Count[j] = 0;
   3. For i = 1 to n do
   4.   For k = 1 to 3 do
   5.      For j = i downto 1 do

Q1b. Compute the asymptotic time complexity in the above code fragment in terms of n. [5]

Q1b. Explain the asymptotic space complexity in the above code fragment. [5]

Q1c. Explain the asymptotic time complexity in terms of n of the following pseudo-code fragment: [5]
   1. For i = 1 to n do
   2.   For j = 1 to 5 do
   3.       Count[j]++;
Q2. Write a linear algorithm for merging three input list of sorted numbers. [20]
Q3. Write an algorithm for the following problem
INPUT: A list of numbers \( L \), and a key \( k \) from that list \( L \).
OUTPUT: A list of numbers \( O \), such that all numbers are from \( L \), \( O[j] = k \), and for all \( i \),
\( i < j \) implies \( O[i] \leq k \), and \( i > j \) implies \( O[i] \geq k \). \[ 20 \]
Q4a. Can *Depth First Traversal* on a finite graph produce a spanning tree?

Q4b. Is it possible that the output $T$ of a minimum-spanning-tree algorithm for an input weighted undirected graph $G$ has less than $n-1$ number of arcs in $T$? Explain your answer very briefly.

Q4c. Does each iteration of the quicksort algorithm divides the input problem into two equal halves?

Q4d. In a city highway system we would like to know what is the shortest distance for connecting all addresses by a network without having any duplicate path between any pairs of addresses. Mention how the problem will be solved. You need not write the algorithm.

Q4e. Explain briefly the following statement: A problem $P$ is in NP-class, but neither NP-complete nor in P-class.
Q5a. Show the computation of the optimal profit for a 0-1 knapsack problem, with a knapsack of limit 9kg, by filling the following table corresponding to the Dynamic Programming algorithm. The following is the input list of objects.
\{O1(5kg, $3), O2(3kg, $30), O3(6kg, $14), O4(8kg, $47)\}. [20]

*To remind you the recurrence: \( P(n,m) = \max\{P(n-1, m), p_n + P(n-1, m - w_n)\} \), for \( w_n > m \), Otherwise \( P(n,m) = P(n-1, m) \). Initialization: \( P(n,m)=0 \) for \( n=0 \), or \( m=0 \).*

<table>
<thead>
<tr>
<th>( w )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ }</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>{O₁}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>{O₁, O₂}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>{O₁, O₂, O₃}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>