1. Consider three keys $K_1$, $K_2$, and $K_3$, such that $K_1 < K_2 < K_3$. Two of the possible binary search trees, $BST_1$ and $BST_2$, are shown below:

```
BST1
K3
  K2
   K1

BST2
K3
  K2
   K1
```

Let $p_i$ denote the probability of retrieving key $K_i$, $1 \leq i \leq 3$. Assume $p_2 = 0.3$ and all key retrievals are successful, that is, the probability of any unsuccessful retrieval is zero. Is it possible that $BST_2$ is better than $BST_1$ in terms of the average number of key comparisons for a retrieval? If the answer is yes, specify the condition for the situation to happen. [14%]

2. A descending priority queue is a data structure into which new items can be inserted in an arbitrary order and from which each deletion-removes the item with the largest value in the information field. Consider a descending priority queue whose information field consists of an integer. Compare the efficiency, in terms of time complexity, of the implementations of the descending priority queue using the following two different types of linked lists: unordered linked list and ordered linked list. [13%]

3. (a) Describe the fundamental concept of hashing technique. [7%]
   (b) Given a sequence of keys as follows: 18, 28, 36, 75, 15, 26, 47, 24, and a division hash function $f(x) = x \mod 9$, show the resulting hash table provided that linear open addressing technique is used to handle overflow and each bucket has one slot. [8%]

4. Suppose that QuickSort is used to sort a file into nondecreasing order. Analyze the time complexity of QuickSort if the file is originally in nonincreasing order. [8%]

5. Show how to implement three stacks in one array. [8%]

6. A full node is a node with two children. Prove that the number of full nodes plus one is equal to the number of leaves in a binary tree. [10%]

7. (a) Construct a Huffman tree from the following set of frequencies. (Always put the smaller subtree to the left. If there is a tie, put the leaf to the left.) [8%]

<table>
<thead>
<tr>
<th>Letters</th>
<th>a</th>
<th>c</th>
<th>e</th>
<th>i</th>
<th>o</th>
<th>n</th>
<th>s</th>
<th>t</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequencies</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

   (b) Encode the following message, based upon the Huffman tree in part (a).
   Message: betitasitis [8%]

8. (a) Prove that any comparison-based algorithm to sort 5 elements requires 7 comparisons. [8%]
   (b) Give an algorithm to sort 5 elements with 7 comparisons. [8%]