1. (Continuation of Problem from Last Assignment)) Consider a Binary Search Tree \( T \) with vertices \( a, b, c, d, e, f, g, h \) and \( \text{ROOT}[T] = a \) and with the following values (\( N \) means NIL)

<table>
<thead>
<tr>
<th>vertex</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
<td>N</td>
<td>e</td>
<td>e</td>
<td>a</td>
<td>d</td>
<td>g</td>
<td>c</td>
<td>a</td>
</tr>
<tr>
<td>left</td>
<td>h</td>
<td>N</td>
<td>N</td>
<td>c</td>
<td>N</td>
<td>f</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>right</td>
<td>d</td>
<td>N</td>
<td>g</td>
<td>N</td>
<td>b</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

(a) Which is the successor of \( c \). Illustrate how the program \textsc{SUCCESSOR} will find it.

(b) Which is the minimal element? Illustrate how the program \textsc{MIN} will find it.

(c) Illustrate the program \textsc{DELETE}[e]

2. Draw binary search trees of height 2,3,4,5,6 on the set of keys \{1,4,5,10,16,17,21\}.

3. What is the difference between the binary-search property and the heap property? (*) Can the heap property be used to print out the keys of an \( n \)-node tree in sorted order in \( O(n) \) time? Explain how or why not.

4. You are given an array \( A[1 \cdots n] \), whose values come from a universe \( \Omega \). (In application, the values would be the keys of records.) You want to test if there are any duplicates, if there are any \( 1 \leq i < j \leq n \) such that \( A[i] = A[j] \). You are given a hash function \( h : \Omega \to \{1, \ldots, n\} \) and a table \( T[1 \cdots n] \) of linked lists, initially all empty. Using the hash function, give an algorithm that returns \textsc{BAD} if there is a duplicate and \textsc{GOOD} if there is no duplicate. Discuss the time of the algorithm under the assumption that calculating the hash function takes unit time.
5. What would the BST tree look like if you start with the root $a_1$ with $key[a_1] = 1$ (and nothing else) and then you apply

$\text{INSERT}[a_2], \ldots, \text{INSERT}[a_n]$ 

in that order where $key[a_i] = i$ for each $2 \leq i \leq n$? Suppose the same assumptions of starting with $a_1$ and the key values but the INSERT commands were done in reverse order

$\text{INSERT}[a_n], \ldots, \text{INSERT}[a_2]$ 

He is a down-to-earth mathematician. There are no stuffy definitions and high theories in his language, just pictures scratched here and there, and some stray Greek letters as mnemonics. He cannot think sitting in a room. If you ask him a question, he will repeat the question, check that he understood the question correctly. He will apologise for being slow; he will say that he cannot understand anything these days, but that he “did understand” the question you posed. Then, he will disappear for a long walk, in the snow and rain outside. When you meet him next, if he has the answer, he will start scratching on the board apologising again and again for his bad handwriting. He will make a claim, and then you must tell him why the claim might be false (that is, what you think the counter example might look like); then he will say, oh! that cannot happen because of this.... Then, you will find another reason why it might be false, and he will tell you why that can’t happen either, and this will go on. At some point, you give up. He will then say that everything needs to be checked. He is a careful mathematician.

– Jaikumar Radhakrishnan, on Endre Szemerédi, 2012 Abel Prize Laureate