Fundamental Algorithms, Assignment 5
Due Monday, March 2, 8:00 a.m. via Gradescope

What we need is more people who specialize in the impossible.
– Theodore Roethke

1. Consider hashing with chaining using as hash function the sum of the numerical values of the letters (A=1,B=2,...,Z=26) mod 7. For example, \( h(\text{JOE}) = 10+15+5 \mod 7 = 2 \). Starting with an empty table apply the following operations. Show the state of the hash table after each one. (In the case of Search tell what places were examined and in what order.)
   - Insert COBB
   - Insert RUTH
   - Insert ROSE
   - Search BUZ
   - Insert DOC
   - Delete COBB

2. 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>e</td>
<td>c</td>
<td>N</td>
<td>f</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>
<tr>
<td>key</td>
<td>80</td>
<td>170</td>
<td>140</td>
<td>200</td>
<td>150</td>
<td>143</td>
<td>148</td>
<td>70</td>
</tr>
</tbody>
</table>

Draw a nice picture of the tree. Illustrate \( \text{INSERT}[i] \) where \( \text{key}[i]=100 \).

3. Continuing with the Binary Search Tree of the previous problem:
   (a) Which is the successor of \( c \). Illustrate how the program \( \text{SUCCESSOR} \) will find it.
   (b) Which is the minimal element? Illustrate how the program \( \text{MIN} \) will find it.
   (c) Illustrate the program \( \text{DELETE}[e] \)

4. 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 \( \text{key}[a_i] = i \) 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

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

5. Set \( N = 2^K \). We’ll represent integers \( 0 \leq x < N \) by \( A[0 \cdots (K - 1)] \) with \( x = \sum_{i=0}^{K-1} A[i]2^i \). (This is the standard binary representation of \( x \), read right to left.) Consider the following algorithms:

Procedure \( \text{JACK}[A] \)

\[
\begin{align*}
I & \leftarrow 0 \\
A[0] & \leftarrow + + \\
\text{WHILE (} & (A[I] = 2 \text{ AND } I < K - 1) \\
\quad & A[I] \leftarrow 0 \\
\quad & I \leftarrow + + \\
\quad & A[I] \leftarrow + + \\
\text{END WHILE}
\end{align*}
\]

and:

\( \text{ANYA}[A] \)

\[
\text{FOR } J = 1 \text{ TO } N - 1 \text{ DO } \text{JACK}[A] \text{ END FOR}
\]

(a) If the input to \( \text{JACK}[A] \) is the binary representation of \( x \) with \( 0 \leq x \leq N - 2 \) describe what the output (the final value of \( A \)) will be.

(b) For “time” we will mean here the number of times the line: “WHILE (\( A[I] = 2 \text{ AND } I < K - 1 \))” is reached. We want here the “time” as a function of \( N \). What is the worst-case time for \( \text{JACK} \)? What is the best-case time for \( \text{JACK} \)?

(c) Assume the array \( A \) is initially all zeroes. Describe what \( \text{ANY}A \) is doing.

(d) (*) Again assume the array \( A \) is initially all zeroes and “time” as above. What is the time for \( \text{ANY}A \) in \( \Theta \)-land?

We don’t yet know how to teach machines to lie.

Ian McEwan, Machines Like Me