Basic Algorithms, Problem Set 6

Due by 8 a.m. Wednesdy, March 10.
Send to Jingshuai: jj2903@nyu.edu.

We don't yet know how to teach machines to lie.
Ian McEwan, Machines Like Me

1. Consider a Binary Search Tree T with vertices a, b, c, d, e, f, g, h and $ROOT[T] = a$ and with the following values (N means NIL)

vertex	a	b	c	d	e	f	g	h
parent	N	e	e	a	d	g	c	a
left	h	N	N	e	c	N	f	N
right	d	N	g	N	b	N	N	N
key	80	170	140	200	150	143	148	70

Draw a nice picture of the tree. Illustrate `INSERT[i]` where `key[i]`=100.

2. Continuing with the Binary Search Tree of the previous problem:

   (a) Which is the successor of c. Illustrate how the program `SUCCESSOR` will find it.

   (b) Which is the minimal element? Illustrate how the program `MIN` will find it.

   (c) Illustrate the program `DELETE[e]`

3. 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

$$INSERT[a_2], \ldots, INSERT[a_n]$$

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

$$INSERT[a_n], \ldots, INSERT[a_2]$$

4. Set $N = 2^K$. We'll represent [1] integers $0 \le x < N$ 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:

[1] Representing integers as binary strings is a powerful idea that comes up frequently.

Procedure $JACK[A]$
$I \leftarrow 0$
$A[0] + +$
WHILE ($A[I] = 2$ AND $I < K - 1$)
    $A[I] \leftarrow 0$
    $I + +$
    $A[I] + +$
END WHILE

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

5. Let T be a binary search tree on nodes $1, \ldots, N$ (in no particular order in the tree) with height H For any vertex v define $depth[v]$ as the distance from v to the root. (The root has depth zero, its children have depth one, grandchildren two, etc.) Let TD denote the sum of $depth[v]$ for all nodes v.

   (a) Give an algorithm to find any particular $depth[i]$ in time $O(H)$ and TD in time $O(HN)$.

   (b) Modifying In-Order-Tree-Walk give an algorithm that finds *all* $depth[i]$ and also TD in total time $O(N)$ – regardless of the value of H.

One of the assumptions we have in liberal democracies is that human beings are intrinsically of value, that they have a value that is not conditional on what they can contribute to the larger society or to the economy or to some sort of common project. If it starts to look like we can be reduced to the point where we're just a bunch of algorithms, I think that seriously erodes the idea that each person is unique and therefore worthy of respect and care regardless of what they can or can't contribute to our joint enterprise.
Kazuo Ishiguro, Klara and the Sun[2]

[2]Ishiguro is a great writer. This new novel has an AI theme. Highly recommended!