Algorithms I (CS21003)

Autumn 2010

Assignment 3

A3. (August 10, 2010)

(a) Use the code of A2 to dynamically allocate a list L of n *positive* integers and randomly assign the elements in L such that each element lies in the closed interval $[a, b]$. Print the input list L.

[User input: n, a, b]

Output: File "`roll number_a30.txt`"

(b) Construct a BST (binary search tree) T_1 using the elements of L one by one.

Each node of T_1 should contain its key and level ($= 0$ implies root).

Report the keys of T_1 in two forms:

i) nodes are reported level-wise, from left to right along each level.

Suggestion: Queue as a data structure.

Output: File "`roll number_a31.txt`" (with '-1' at the start and 'new line' at the end of each level)

ii) by in-order traversal.

Output: File "`roll number_a32.txt`" (append max level)

(c) Write a procedure to search a key in T_1.

Hence compute the average number of comparisons for searching all elements of L in T_1, one at a time.

Append this figure in "`roll number_a32.txt`".

(d) Compute the average number of comparisons for searching all elements of $[a, b] \setminus L$ in T_1, one at a time.

Append this figure in "`roll number_a32.txt`".

(e) Write a procedure to search all elements in T_1 lying the user-specified interval, $[c, d]$.

Append this figure in "`roll number_a32.txt`".

(f) Find recursively the median (using A2) of L and reconstruct a BST, T_2, such that its height is $\lceil \log_2 n \rceil$.

Perform the above operations (b–e) on T_2 and report the output data in "`roll number_a33.txt`" and "`roll number_a34.txt`", in the same fashion as mentioned above.

(g) Construct a height-balanced BST (AVL tree), T_3, from L.

Perform the above operations (b–e) on T_3 and report the output data in "`roll number_a34.txt`" and "`roll number_a35.txt`", again in the above-mentioned fashion.