1994 Paper 2 Question 9

Recall that a *binary search tree* is a binary tree whose labels are ordered from left to right, while a *priority queue* is a collection whose elements are inserted in any order but are removed in increasing order. Priority queues can be represented by binary search trees using the normal insert operation; they require additional functions *least* and *del_least*. Given a binary search tree, *least* returns the node with the smallest label, while *del_least* returns the binary search tree with that node removed.

Using a suitable `datatype` for binary trees, code *del_least* as a recursive (and side-effect free) ML function. Illustrate its operation using an example. [4 marks]

Consider now the cost of emptying an n-label binary search tree by applying *del_least* to it n times. The cost will be measured by the number of calls to *del_least*, including recursive calls.

Which binary search trees constitute the best and worst cases for this operation? State the cost of each as a function of n. [4 marks]

If the tree is perfectly balanced and of depth d then $n = 2^d - 1$. Show that the cost of emptying the tree in this case is $d \times 2^{d-1}$. [8 marks]

A functional program cannot update the binary tree 'in place' but must do some copying. To what extent does the number of calls to *del_least* reflect the true cost of emptying the tree? Discuss. [4 marks]