2000 Paper 1 Question 5

Foundations of Computer Science

This question has been translated from Standard ML to OCaml

State the time complexity of the *lookup* and *update* operations for each of the following:

(*a*) association lists

(*b*) binary search trees

(*c*) functional arrays (implemented as binary trees)

Use O-notation and include both the average-case and worst-case complexity. [6 marks]

You are provided with the OCaml code for binary search trees, including the *lookup* and *update* operations. Use these operations to code a sorting function that works by repeatedly inserting elements of a list into a binary search tree, then converting the final binary search tree back into a list. [4 marks]

Consider the following methods of sorting a list:

(*a*) Locate the smallest element of the input. The output is this element followed by the result of recursively sorting the remaining elements.

(*b*) Take the first 16 elements of the input and sort them using special hardware. Sort the remaining elements recursively. The output is the result of merging the two sorted lists.

(*c*) Take the first 20% of the input elements. Sort them and the remaining 80% recursively. The output is the result of merging the two sorted lists.

For each of these methods, state with justification the worst-case complexity in terms of the number of comparisons. [10 marks]