The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 7, Problem 7.40

We know (see Exercise 7.24) that the associated decision problem is NP-complete; thus we need only show that the optimization version is NP-easy.

We show how to use an oracle for the decision problem to solve the optimization problem. Notice that the external path length is an integer quantity and cannot exceed a maximum (using only type $\{x_i\}$ tests) of $n + 2 + \Sigma_i i,\ 1 \leq i \leq n + 2 = (n^2 + 7n + 10)/2$; thus we can use the oracle in a binary search to determine the minimum cost for the problem with at most $O(\log n)$ queries. Then we can undertake the process of finding the root: pick an arbitrary test and separately find the minimum cost for each of the 2 subproblems (using the same binary search procedure), all of which can be done in time $O(n)$; then put these costs together to get the cost of the optimal tree with that root choice (just the sum of the 2 costs and of the number of objects in the problem at hand) and compare it with the known optimal; if equal, we have found the root, if not, try another test. The process is then repeated recursively on each subproblem. The total time is at worst $O(n^2)$ for a root, or $O(n^3)$ for the whole tree, with at most $O(n^2 \cdot \log n)$ queries to the oracle. Thus the optimization problem is NP-easy; since it is obviously NP-hard, we are done.