Design of Algorithms - Homework III

K. Subramani
LCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Instructions

(i) The homework is due on April 4, in class.

(ii) Attempt as many problems as you can. You will get partial credit, as per the policy discussed in class.

2 Problems

1. Professor Krustowski claims to have discovered a new sorting algorithm. Given an array $\mathbf{A}$ of n numbers, his algorithm breaks the array into 3 equal parts of size $\frac{n}{3}$, viz., the first third, the middle third and the bottom third. It then recursively sorts the first two-thirds of the array, the bottom two-thirds of the array and finally the first two-thirds of the array again. Using mathematical induction, prove that the Professor has indeed discovered a correct sorting algorithm. You may assume the following: The input size n is always some multiple of 3; additionally, the algorithm sorts by brute-force, when n is exactly 3. Formulate a recurrence relation to describe the complexity of Professor Krustowski's algorithm and obtain tight asymptotic bounds.

2. Assume that you are given a chain of matrices $\langle A_1 \ A_2 \ A_3 \ A_4 \rangle$, with dimensions 2×5, 5×4, 4×2 and 2×4 respectively. Compute the optimal number of multiplications required to calculate the chain product and also indicate what the optimal order of multiplication should be using parentheses.

3. A hiker has a choice of n objects $\{o_1, o_2, \ldots, o_n\}$ to fill a knapsack of capacity W. Object o_i has benefit p_i and weight w_i. A subset of objects is said to be feasible, if the combined weight of the objects in the subset is at most W. The hiker's goal is to select a feasible subset of objects, such that the benefit to him is maximized (benefits are additive). Note that an object cannot be selected fractionally; it is either selected or not. Design a dynamic program to help the hiker.

4. Let T denote a binary search tree. Show that
   (a) If node a in T has two children, then its successor has no left child and its predecessor has no right child.
   (b) If the keys in T are distinct and x is a leaf node and y is x's parent, then $y \cdot key$ is either the smallest key in T larger than $x \cdot key$, or the largest key in T smaller than $x \cdot key$.

5. An AVL tree is a binary search tree that is height balanced: for each node x, the heights of the left and subtrees of x differ by at most 1. Prove that an AVL tree with n nodes has height $O(\log n)$.