ESO207 Programming Assignment-2

Instructions

- Please insert suitable comments in your pseudo-code and actual code so that anyone grading it may understand it easily.

- Present your arguments clearly and stay to the point.

- Test cases in hackerrank may not have been designed to evaluate complexity of your algorithms. These are only to help see if you have a working program.

- Runtime complexity of your program may be assessed from complexity analysis of your pseudo-code and by checking the fact that you have implemented that pseudo-code.

- Your code is allowed to be destructive. That is, input arguments may not denote the same object after the computation.

- No marks shall be awarded for algorithms/programs not working in the prescribed theoretical time bounds.

This assignment is about implementing some operations of Union-Find-Split ADT used in algorithm to find largest common subsequence. Each set is represented as a 2-3 tree. For examples of these operations, you may refer to Lecture 23.

Height of a tree T is denoted by $h(T)$. For a set S, $|S|$ stands for number of elements in S.

Q1 (Marks 15 + 20) You are given 2-3 trees T_1 and T_2, representing respectively finite sets S_1, S_2 of natural numbers. Further, it is given that for all $x \in S_1$ and for all $y \in S_2$, $x < y$.

    (a) Write the pseudo-code for an algorithm to compute 2-3 tree representation of $S = S_1 \cup S_2$. Your algorithm should take $O(h(T_1) + h(T_2))$ time. Justify time complexity of your algorithm. Name your function as $Merge(S_1, S_2, S)$.

    (b) Implement your algorithm of part (a) as an actual program in hackerrank.

Q2 (Marks 35 + 30)

    (a) Suppose you are given a 2-3 tree T representing a set S of natural numbers and a natural number x. Write the pseudo-code for an algorithm to split T into two 2-3 trees T_1, T_2 representing sets S_1, S_2 respectively, where
$S_1 = \{y \in S \mid y < x\}$ and
$S_2 = \{y \in S \mid y \geq x\}$.
Your algorithm should take $O(h(T))$ time. Do the complexity analysis of your pseudo-code to prove this bound. Name your function as $Split(T, x, T_1, T_2)$.

    (b) Implement your algorithm of part (a) as an actual program in hackerrank.

——— End of Assignment ———