

University of Houston

COSC 3320: Algorithms and Data Structures
Summer 2015

Homework 3

Due June 24, at the start of class

1. Given an array $A[1, 2, \dots, n]$ of n elements, a *majority* element of A is an element occurring at least $\lceil (n+1)/2 \rceil$ times. The elements cannot be ordered or sorted, but can be compared for equality. Design an efficient divide and conquer algorithm that returns a majority element of A (if any), and determine its complexity.
2. Design and analyze an algorithm `preorderNext(T, v)` that, given a binary tree T and a node $v \in T$, returns the node visited immediately after v in the preorder visit of T (and returns `null` if v is the last node visited in the preorder visit of T).
3. Let T be a proper binary tree. Define the *heightsum* of T as the sum of all the heights of the nodes of T .
 - (a) Determine an upper bound to the heightsum of a proper binary tree with n nodes, and describe a tree whose heightsum is such a value.
 - (b) Design a divide and conquer algorithm `heightSum(T, v)` that computes the heightsum of T_v , where T_v denotes the subtree of T rooted at $v \in T$.
 - (c) Analyze the complexity of `heightSum(T, T.root())`.