Assignment 1, ADS I1, Autumn 2015

Your solution of this assignment must be handed in before Friday 02-Oct-2015 at 23:59. It will be a part of the evaluation of the course. If you fail this assignment you will not be allowed to attend the exam. At the exam you have to talk about one of your assignments.

The assignment is individual. Similar solutions from different persons will not be accepted.

In this assignment you have to design and implement a “Hit Balanced Tree” (HBT). HBT is a new data structure invented for this assignment and it might not be of any use in real programs.

An HBT is an ordinary binary search tree where all the nodes contain an extra attribute – number of hits. When a node is created the hit count is set to zero. Every time the node is found with the find(), contains() etc. methods its hit count is incremented by one.

When an HBT is balanced, all the descendants of a node have either the same or a lower hit count than the node itself. On the other hand, it is still a search tree, where all nodes in the left subtree of node have elements smaller than the element of the node, and corresponding greater than for the right subtree.

In other words: the root will be the node with the highest hit count. To the left are the smaller elements and the larger element are to the right. This rule is applied recursively down both subtrees.

Notice that a HBT is *not* balanced in the classical sense. A HBT doesn’t guarantee an average search time of O(log n).

Implement an HBT and make a proper test of your implementation.