CS2302
Fall 2011
Lab 4
B-trees and Hash Tables
Due electronically by 11:59 p.m., October 24, 2011

Instructions

In this lab you will compare the efficiency of binary search trees, B-trees, and hash tables with chaining for search and storage.

You will generate semi-random sequences of integers and then determine the integer that appears the largest number of times in the sequence, breaking ties arbitrarily.

Each experiment to be performed is explained by the following pseudocode:

Generate array A[0],...,A[n-1] of random integers in the range 0 to 100,000. For i = 0 to n-1
- search for A[i] in the data structure (BST, B-tree, or hash table)
- if A[i] is present, increase its counter by one
- else create a new counter for A[i], initialize it to 1 and insert object containing A[i] and its counter to data structure
Find the element A[m] with the largest value of counter and display A[m] and its number of occurrences

Perform experiments with different values of n (for example, 100000, 1000000, and 10000000) and compare the times it takes to process the arrays with each of the implementations. Use the hash function h(k) = abs(k) % tableSize, where tableSize is the largest prime number that is less than n (write a method to find that number). For the B-tree, use t = 11.

You are not allowed to use the built-in implementations of these data structures; you have to implement your own. Run experiments to compare the running times of your implementations for input files of various sizes and, as usual, write a report explaining your work, as described in the syllabus.