

CS2302 - Data Structures

Fall 2017

Lab # 4

B-trees

Deadline: Wednesday, October 18, 11:59 p.m.

Download the code that implements B-trees and extend it in the following ways:

1. Modify the insertion operation to prevent duplicate elements from being inserted in the tree. If the user tries to insert an element that is already in the tree, your program should simply ignore the request.
2. The search operation inside a node discussed in class is sequential (the given key k is compared to all the elements in the *key* array in the node until one is found that is larger than k , or there are no more keys left to compare). If the number of keys in the node is large, this is inefficient. Modify the search method to perform binary search on the *key* array instead.
3. Add a variable to the BTree class to store the number of keys in the tree and make sure the variable is kept up to date after every insertion operation (observe how the height of the tree is kept in a variable for efficiency, but it could also be easily computed).
4. Add a variable to the BTree class to store the number of nodes in the tree and make sure the variable is kept up to date after every insertion operation.
5. Write methods that perform the following operations:
 - (a) Print the keys in the tree in ascending order.
 - (b) Print the keys in the tree that have depth d in descending order.
 - (c) Determine if a given element k is in the tree.
 - (d) Return the minimum element in the tree.
 - (e) Return the minimum element in the tree at a given depth d .
 - (f) Return the maximum element in the tree.
 - (g) Return the maximum element in the tree at a given depth d .
 - (h) Return the number of nodes in the tree.
 - (i) Return the number of keys in the tree.
 - (j) Return the number of keys in the tree at a given depth d .
 - (k) Return the sum of all the keys in the tree.
 - (l) Return the sum of all the keys in the tree at a given depth d .
 - (m) Return the number of leaves in the tree.
 - (n) Return the number of nodes in the tree that have depth d .
 - (o) Return the number of nodes in the tree that are full.
 - (p) Given a key k , return the depth at which it is found in the tree, of -1 if k is not in the tree.
 - (q) Given a key k , print all the keys that are in the same node as k .

As usual, write a report describing your work.