Data Structures and Algorithms in Python

Michael T. Goodrich
Department of Computer Science
University of California, Irvine

Roberto Tamassia
Department of Computer Science
Brown University

Michael H. Goldwasser
Department of Mathematics and Computer Science
Saint Louis University

Study Guide: Hints to Exercises

WILEY

Chapter 15 Memory Management and B-Trees

Hints

Reinforcement

R-15.1) Perform an Internet search to determine a good estimate on the number of atoms on earth.

R-15.2) Start with the description provided in the book.

R-15.3) Revisit the definition of an (a,b) tree.

R-15.4) The definition of an order-d deals with the minimum and maximum number of children an internal node can have. Please see the book for details.

R-15.5) Draw the memory cache and manually process the requests using a pencil with a good eraser.

R-15.6) Draw the memory cache and manually process the requests using a pencil with a good eraser.

R-15.7) Draw the memory cache and manually process the requests using a pencil with a good eraser.

R-15.8) Use a pencil with a good eraser.

Creativity

C-15.9) Review the external-memory sorting algorithm.

C-15.10) Keep the top one or two blocks of the stack in main memory.

C-15.11) Keep queue runs in blocks.

C-15.12) Consider an alternate linked list implementation that uses "fat" nodes.

C-15.13) Note that each valid node v and its children in a (2,4) tree correspond to a red-black subtree of height 2. In a (4,8) tree, you will need bigger subtrees.

C-15.14) Consider the extreme cases.

C-15.15) Try to block order-B sized sub “trees” in the skip list.

C-15.16) Start from sequence solution for the union-find problem.

C-15.17) A single scan suffices.

C-15.18) Each request can “see into the future” to see when is the next time existing blocks will be accessed next.

C-15.19) In an initial scan, keep track of the best candidate majority value, x, and a counter that keeps track of the number of times you have seen a copy of x versus some other integer.

C-15.20) Consider what happens to a page that is accessed a lot and then never accessed again.

C-15.21) The answer just uses some simple logarithm identities.

Projects

P-15.22) Make sure to use typical memory sizes and do a long simulation.

P-15.23) Let a and b be definable parameters or constants. And let insertion be the first update method you program.

P-15.24) Start with insertion as the first update operation you code up, and use a simple uniform distribution of keys to perform the experiments.