

*Institut für Theoretische Informatik**Prof. Dr. Peter Widmayer, Dr. Riko Jacob and Yoshio Okamoto**November 4, 2004*

External Memory Algorithms and Data Structures Problem 3

Course Webpage: <http://www.ti.inf.ethz.ch/ew/courses/EMADS04/>

Topic: B-trees

Due Date: November 11, 2004 at the lecture

Exercise 3.1

Design an external-memory linked list data structure for which we can perform an insertion and a deletion in $O(1)$ IOs and a search in $O(N/B)$ IOs.

Exercise 3.2

- We are given two $(2, 4)$ -trees T' and T'' storing some distinct natural numbers, and a natural number x which is larger than all numbers in T' and smaller than all numbers in T'' . We want to construct a $(2, 4)$ -tree for x and the numbers in T' and T'' . (This operation is called the *join*.) Design an algorithm to perform the join in $O(1 + |h' - h''|)$ time where h' and h'' are the heights of T' and T'' respectively.
- We are given a $(2, 4)$ -tree T storing n distinct natural numbers, and a number x in T . We want to construct two $(2, 4)$ -trees T' consisting of all numbers in T smaller than x and T'' consisting of all numbers in T larger than x . (This operation is called the *split*, and it is a kind of “converse” of a join.) Design an algorithm to perform the split in $O(\log n)$ time. (Hint: Consider a path in T from the root to x . This path breaks the numbers smaller than x in T into some trees. Then apply the join operations.)

Exercise 3.3

We would like to perform a *finger search* in an external B-tree; Namely given a leaf v of the B-tree, search for another leaf w by utilizing the closeness of v and w . By extending B-trees, design a data structure and an algorithm to do this task with $O(\log_B Q)$ IOs. Here, Q is the number of leaves between v and w .