Uni Freiburg, Web Science Group
Prof. Peter Fischer
Systems Infrastructure for Data Science - Winter 2014/15

Exercise Sheet #2: Indexing

Due/discussion: Nov 6, 2014

Exercise 2.1 : B$^+$ Trees

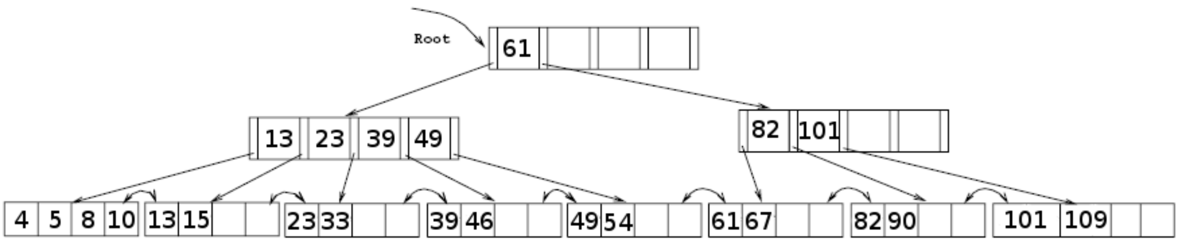

Figure 1: B$^+$ Tree with d=2

Consider the B$^+$ Tree shown in Figure 1 for the following questions. Apply the operations to the *original* tree shown in the figure and show the situation of the tree after the operations. Assume at least 50% occupancy. In case of redistribution, use the left sibling unless otherwise stated.

A. Insert an entry with key 14.

B. Insert an entry with key 6.

C. Delete the entry with key 13.

D. Repeat C, assuming that *right* sibling is checked for possible redistribution.

E. Insert an entry with key 56 and then delete the entry with key 61.

F. Insert an entry with key 70 and then delete the entry with key 101.

G. Delete the entries with the following keys in that order: 39, 46, 49, 54, 82.

H. **Bulk loading:** Assume a B$^+$ Tree with order d=1. Bulk load the tree with the following key values making sure that each leaf is full : 2, 4, 6, 8, 10, 12, 14, 16.

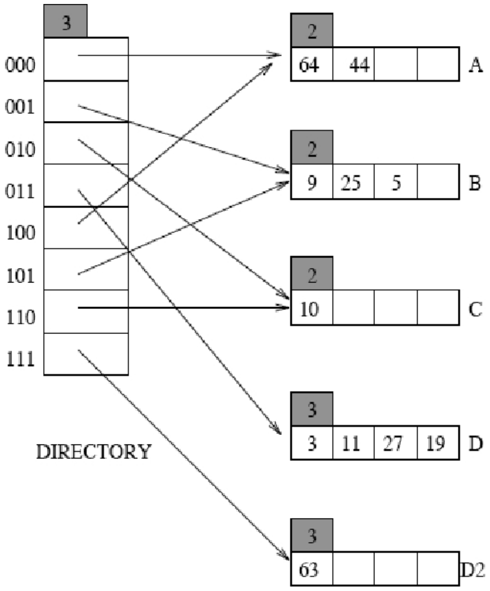

Figure 2: Extendible Hashing Index

Exercise 2.2 : Hash-based Indexing

A. Consider the Extendible Hashing index shown in Figure 2. Explain what happens when the following operations are performed in the given order: 1) Insert 13 2) Insert 17 3) Delete 63

B. Consider the Linear Hashing index shown in Figure 3 where *Level* = 0 and $N = 4$. Explain what happens when we insert entries: 43, 37, 29, 22, 66, 34, 50

After completing these operations, explain what happens if we delete entries 7 and 31.

C. Give an example of a Linear Hashing index and an Extendible Hashing index which have the same entries, such that the Extendible Hashing index has more pages than the Linear Hashing index. Assume the entries are inserted in the same order for both indexes and that a bucket can hold 4 entries.

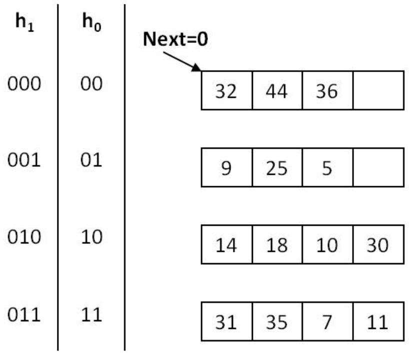

Figure 3: Linear Hashing Index