CS 240 — LECTURE 23

Bartosz Antczak *Instructor: Eric Schost* *March 30, 2017*

23.1 More on B-trees

23.1.1 Height of a B-tree

The height of a tree with n nodes is $\Theta((\log n)/(\log M))$. For the exam, know the algorithms that can be applied on this tree (delete, search, insert), and also know that the height of a tree is $\Theta((\log n)/(\log M))$.

Proof of the height of a B-tree

M is the maximum number of children for any node. With that in mind, observe that the number of keys n in the tree is

$$n \geq \left(\frac{M}{2}\right)^h$$
$$\log(n) \geq h \log\left(\frac{M}{2}\right)$$
$$\frac{\log(n)}{\log(\frac{M}{2})} \geq h$$
$$h \in O\left(\frac{\log(n)}{\log(M)}\right)$$

Similarly we have:

$$n \leq M^{h+1}$$
$$\log(n) \leq (h+1)\log(M)$$
$$h \in \Omega\left(\frac{\log(n)}{\log(M)}\right)$$

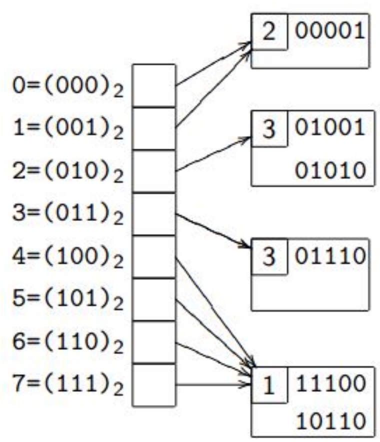
Thus, we have $h \in \Theta\left(\frac{\log(n)}{\log(M)}\right)$

23.1.2 Hashing in External Memory

How can we hash and minimize disk transfers? We can use **extendible hashing**, which is similar to a B-tree with height 1 and max size S at the leaves.

Structure

The directory is stored in internal memory. It contains an array of size 2^d, where d is called the *order*. Each directory entry points to a block stored in external memory. Each block contains at most S items. Note that many entries can point to the same block. The values in every block in block B agree on leading k_B bits: Know how the data structure is constructed. How to search. How to insert items. FORGET ABOUT DELETE.


0=(000)2
1=(001)2
2=(010)2
3=(011)2
4=(100)2
5=(101)2
6=(110)2
7=(111)2
2 00001
3 01001
01010
3 01110
1 11100
10110