**Quiz Sheet for *Architecture and Implementation of Database Systems***

**Prof. Rudolf Bayer, Ph. D.**
Institut für Informatik
SS 2003

Exercises for **Chapter 1: Files**

1. What must be the block size on a hard drive so that the net output (Nutzleistung) for accessing the hard drive reaches 50%? (Chose for yourself the technical parameters of the hard drive from a recent source.)

**Answer**:

ap = actual performance (*in MB/s*)
tr = transfer rate (= *15 MB/s*)
bs = block size (in *MB*)
bt = block transfer time (in *s*)
at = access time (= 0.01 *s*)

net output = ap / tr
ap = bs / bt *B/(MB/s)* = bs / bt *MB/s*
bt = at + bs / tr *s*

$\Rightarrow$ net output = (bs / ( at + bs/tr )) / tr = bs / (at*tr + bs)
net output be 50%:
0.5 = bs / (at*tr + bs)

$\Rightarrow$ bs = at*tr *MB* = 0.01 *s* * 15 *MB* = 0.15 MB = 153.6 *KB*

2. Determine the general formula for the **net transfer time** in s of a hard drive dependent on the average access time $\alpha$ in ms and the transfer rate $\beta$ in B/s.

**Answer**:

$(\alpha / 1000)$ *s* + $(x / \beta)$ *s*, where x is the number of Bytes transferred

3. UNIX files:

a) What is the maximum size of a UNIX file if the block size is 4 KB?

**Answer**:

Block size is 4 KB and size of an address is 4 B:
⇒ number of addresses per block = 4 KB/4B = 1024

i-Node contains 10 direct block addresses, 1 pointer to each, blocks with single indirect, double indirect and triple indirect addresses (cf. Chapter 1.3):
⇒ max. file size = $(10 + 1024 + 1024^2 + 1024^3) * 4$ KB = 4 TB

b) What is the maximally *addressable* size of a UNIX file using byte addressing
- in 32 bit computers using 31 bits per address, and
- in 64 bit computers using 63 bits?

**Answer**:

32 bit: $2^{31}$ B = 2 GB
64 bit: $2^{63}$ B = $2^{33}$ GB

c) What block size must be chosen for a UNIX file in order to achieve the maximum addressability for both addressing schemes?

**Answer**:

32 bit: $2^{31}$ B = $(10 + x + x^2 + x^3) * 4$ KB
⇒ $2^{19} - 10 = x + x^2 + x^3$
⇒ $x \approx 80$ (addresses per block)
⇒ block size is 80 * 4B = 320 B

64 bit: $2^{63}$ B = $(10 + x + x^2 + x^3) * 4$ KB
⇒ $2^{51} - 10 = x + x^2 + x^3$
⇒ $x \approx 2^{17}$ (addresses per block)
⇒ block size is $2^{17} * 4B = 2^{19}B$ = 0.5 MB

4. How may a relation of size 60 GB be mapped on a UNIX file system? Suggest an appropriate tuple identifier.

**Answer**:

With a maximum file size of 2 GB, 30 files are needed to store the relation on the file system (60GB / 2GB = 30).

Space requirement for tuple identifier:
- n bits for addressing a file holding tuples, and
- m bits for specifying the tuples within such a file

n = 5 bits (as $2^5 \geq 30$)
m depends on the tuple size of the relation:

Suppose a relation with 4 attributes of size 4B and 2 attributes of size 8B
$\Rightarrow$ tuple size = 4*4B + 2*8B = 32B
$\Rightarrow$ number of tuples per file = 2GB/32B = $2^{31}/2^5 = 2^{26}$
$\Rightarrow$ m = 26 bits

$\Rightarrow$ total space requirement = n + m = 5 + 26 bits = 31 bits

Exercises for **Chapter 2: B-Trees**

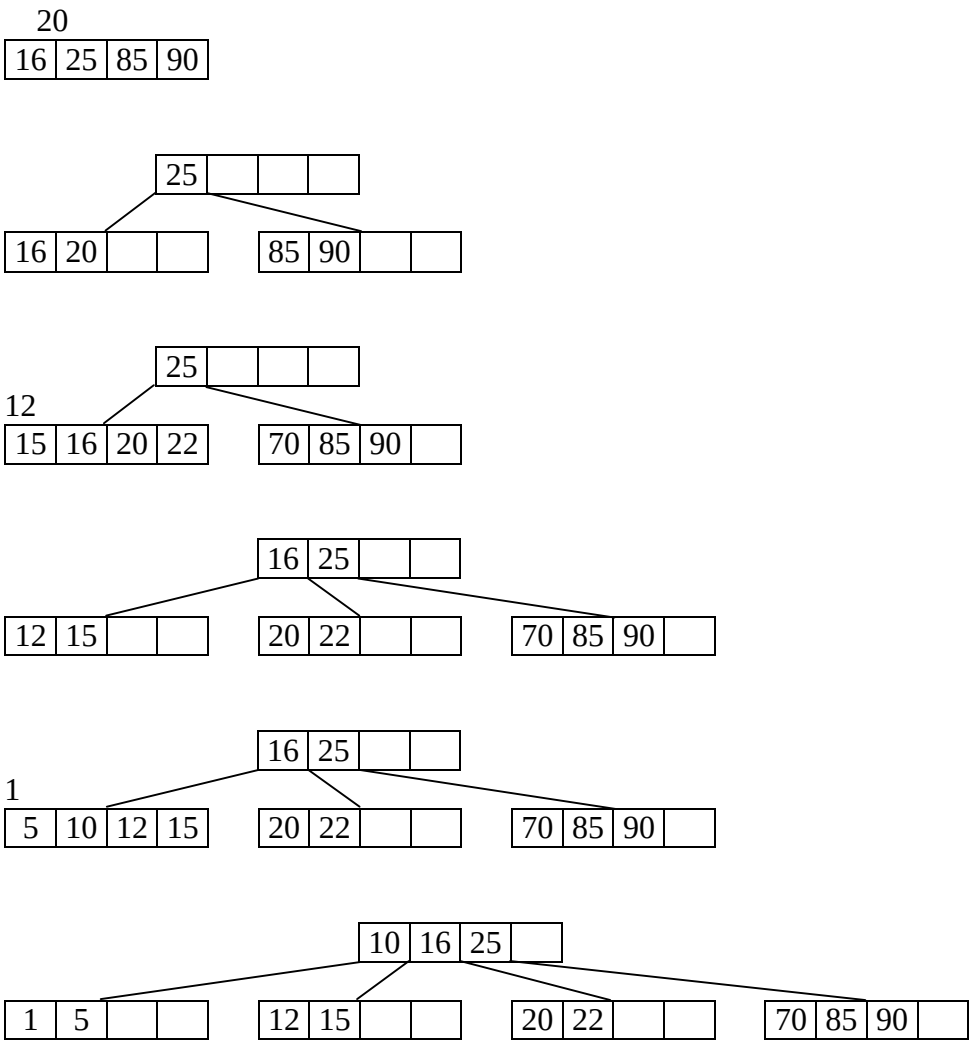
1. How many values can be stored in a B-Tree of order n and depth m, maximally and minimally?

**Answer**:

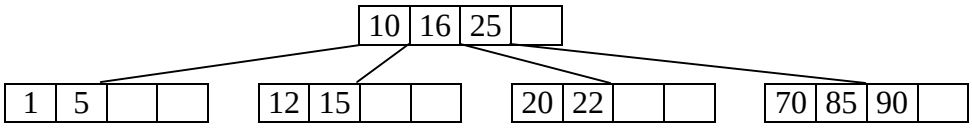
$2n * \sum_{i=0}^{m} (2n+1)^i$ values maximally, and $n * \sum_{i=0}^{m} (n+1)^i$ minimally.

2. Let T be a B-Tree of order 2 and let S = (90, 85, 25, 16, 20, 15, 70, 22, 12, 10, 5, 1) be a sequence of integers that are inserted in this order into T. Sketch the different stages of T in this process.

**Answer**:



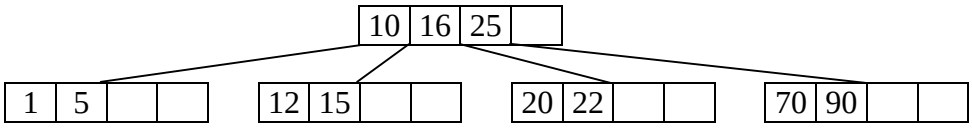
3. Let T be a B-Tree of this form:



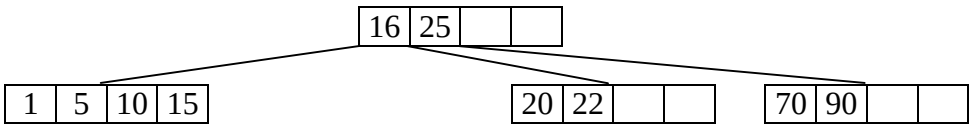

a) Sketch the different stages of the tree when deleting entries 85,12 and 16 (in this order).

**Answer**:

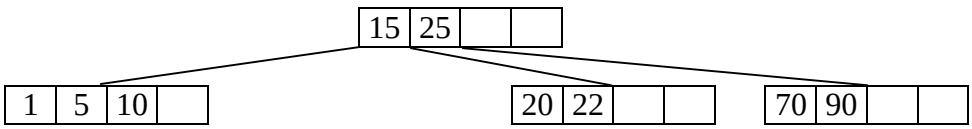
delete 85:




delete 12:




delete 16:




a) How does T change if value 16 is now inserted again?

**Answer**:

insert 16: