Implementation of DBMS
Exercise Sheet 6, Solutions
Klingemann, WS 2024 / 2025

1) We want to represent physical addresses for a hard disk. For a block address we need to identify the following entities: the cylinder, the track within a cylinder, and the block within a track. To each of these entities we allocate one or more bytes to identify it. Our disk has the following properties:

- 8192 cylinders
- 8 tracks in a cylinder
- 32 blocks in a track

a) How many bytes do we need for a block address?

Solution:

- The disk has $8192 = 2^{13}$ cylinders. Thus, we need 13 bits to identify the cylinder. As a byte consists of 8 bits, we require 2 bytes.
- The disk has $8 = 2^3$ tracks in a cylinder. Thus, we need 3 bits to identify the track within the cylinder. This means we require 1 byte.
- We have $32 = 2^5$ blocks per track. Hence, we need 5 bits and therefore assign 1 byte to this part of the address.

As a result, in total we need 2 + 1 + 1 bytes = 4 bytes for a block address.

b) We want construct a record address by adding the position of the byte within a block to the block - address of exercise a). The blocks of the disk consist of 4096 bytes How many bytes would we need for the record address?

Solution:

The blocks of this disk consist of 4096 bytes = 2^{12} bytes. Therefore, we need 12 bits to identify the starting position of the record within the block. This means we require 2 bytes. In total we need 4 + 2 bytes = 6 bytes for the record address.

2) Suppose that blocks can hold either ten records or 100 key-pointer pairs. We have a data file that is a sequential file and a sparse index on this file. The index has multiple levels up to a level with just one block. Each primary block of the data file has one overflow block. The primary blocks are full, and the overflow blocks are half full. However, records are in no particular order within primary block and its overflow block. All index blocks are 60% full.

a) Calculate the total number of blocks needed for a 3,240,000-record file and the index.

Solution:

A pair of a primary block and an overflow block contains 10 + 5 records = 15 records. Therefore, we need for the data file $\lceil$(3,240,000 records) / (15 records / pair of blocks)$\rceil$ = 216000 pairs of a primary block and an overflow block. Hence, we need 432000 blocks.

As we have a sparse index, we need one pointer for each primary block of the data file. As the index blocks can store 100 key-pointer pairs but are only 60% full, we have in an index block 60 key-pointer pairs. Therefore, we need for the index levels:

1st level index: $\lceil$(216000 key-pointer pairs) / (60 key-pointer pairs / block)$\rceil$ = 3600 blocks.

2nd level index $\lceil$(3600 key-pointer pairs) / (60 key-pointer pairs / block)$\rceil$ = 60 blocks

3rd level index $\lceil$(60 key-pointer pairs) / (60 key-pointer pairs / block)$\rceil$ = 1 block

In total we need 3661 blocks for the index and 435661 blocks including the data file.

b) Calculate the average number of disk I/O's needed to retrieve a record given its search key by using the index. You may assume that nothing is in memory initially, and that the search key is the primary key for the records.

Solution:

We need one I/O for each level of the index, i.e., 3 I/O's for the index.

For finding a record in the primary bock, we need to examine just one block. For finding a record in the overflow block we need to examine two blocks. The fraction of records which reside in a primary block is 10/15 = 2/3 and for an overflow block 5/15 = 1/3. Therefore, the average number of I/O's for the data file is: 2/3 * 1 I/O + 1/3 * 2 I/O's = 4/3 I/O's.

In total, the average number of I/O's is 4 + 1/3 I/O's.

3) In a B+-tree (will be discussed later in the lecture) of order n, the minimum number of keys in a node can be calculated with the following formulars:

- non-leaf node: $\lceil (n+1)/2 \rceil - 1$
- leaf node: $\lfloor (n+1)/2 \rfloor$

What is the minimum number of keys in B+-tree (i) interior nodes and (ii) leaves, when

a) n = 10

b) n = 11

Solution:

We just have to insert the value of n into the corresponding formula which gives us:

a) for non-leaf nodes: $\lceil (10 + 1) / 2 \rceil - 1 = \lceil 5.5 \rceil - 1 = 5$
   for leaf nodes: $\lfloor (10 + 1) / 2 \rfloor = \lfloor 5.5 \rfloor = 5$

b) for non-leaf nodes: $\lceil (11 + 1) / 2 \rceil - 1 = \lceil 6 \rceil - 1 = 5$
   for leaf nodes: $\lfloor (11 + 1) / 2 \rfloor = \lfloor 6 \rfloor = 6$