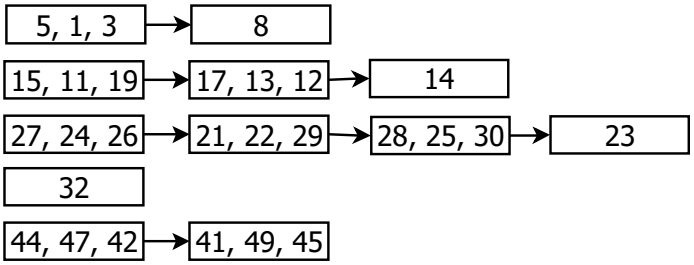
Implementation of DBMS
Exercise Sheet 4, Solutions
Klingemann, WS 2025 / 2026

1) We have a data file with 10^5 records. Records and blocks are like in task 3a) of Sheet 3. How many blocks do we need for the data file?
a) We use spanned storage.
b) We use unspanned storage.
Solution:
a) In total we need 54 bytes / record * 100000 records = 5,400,000 bytes. As we have a block header that consumes 40 bytes, the number of bytes in a block available for storing records is 4096 bytes – 40 bytes = 4056 bytes. Due to spanned storage, we can use each of these bytes and need therefore
⌈5,400,000 bytes / (4056 bytes for records / block)⌉ = 1332 blocks
b) We have already calculated in Sheet 3 that we can store 75 records in each block. Therefore, we need ⌈100000 records / (75 records / block)⌉ = 1334 blocks

2) We have a sequential file that consists of 5 primary blocks. The first block takes records with key values in the range 1-10, the second block takes records with key values in the range 11-20 and so on. Each block can store up to three records. In case of full blocks, we create overflow blocks. Initially all blocks are empty. We insert records with the following key values in the given order:
27, 32, 5, 24, 44, 26, 1, 15, 21, 3, 11, 47, 42, 22, 29, 19, 28, 41, 17, 25, 13, 49, 30, 45, 8, 12, 23, 14
How many overflow blocks does each primary block have?
Solution:
When we perform the insertions, the records are assigned to blocks as indicated below. This means that we get the following overflow blocks:
The first primary block has 1,
the second has 2,
the third has 3,
the fourth has 0,
and the fifth has 1.



3) Suppose that we handle insertions into a sequential data file of n records by creating overflow blocks as needed. Also, suppose that the data blocks are currently all half full. If we insert new records at random, how many records do we have to insert before the average number of data blocks (including overflow blocks if necessary) that we need to examine to find a record with a given key reaches 2? Assume that on a lookup we know in which chain of primary and overflow blocks the record we are looking for is located. We search the primary block first, and only search overflow blocks, in order, until we find the record, which is definitely in exactly one of the blocks of the chain.
Solution:
To calculate the average number of data blocks that we need to examine, we have to take into account which portion of all records in a chain of blocks resides in a particular block. Initially we have just the blocks that we used when creating the file (the primary blocks). When we need more space, we start creating overflow blocks. For example, assume that we have already inserted some records so that for every primary block we have one overflow bock which is half full. Then the portion of records that is in the primary block is 2/3 and for the overflow block 1/3. 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. Therefore, the average value is:
2/3 * 1 examined block + 1/3 * 2 examined blocks = 4/3 examined blocks.
When we continue inserting, we can see that we get an average of 2 when we have for every primary block two completely full overflow blocks as we get in this case:
1/3 * 1 examined block + 1/3 * 2 examined blocks + 1/3 * 3 examined blocks =
2 examined blocks.
As n records mean half full primary blocks, we have in this scenario 6n records. Thus, we added 5n records.

4) 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.