

Photo by [JOSHUA COLEMAN](#) on [Unsplash](#)

Übung Datenbanksysteme II

# Physische Speicherstrukturen

Tobias Bleifuß

Teilweise basierend auf Folien von Leon Bornemann, Maximilian Jenders und Thorsten Papenbrock

# Organisatorisches: Übung Datenbanksysteme II

---

## Übung

- Tobias Bleifuß ([tobias.bleifuss@hpi.de](mailto:tobias.bleifuss@hpi.de))

## Tutoren

- Tom Braun
- Hendrik Patzlaff

## Mailingliste

- [fragen-dbs2-2019@hpi.uni-potsdam.de](mailto:fragen-dbs2-2019@hpi.uni-potsdam.de)
- Hilfe zur Selbsthilfe: Fragen gegenseitig beantworten
- Tutoren und Mitarbeiter lesen mit
- Keine Email erhalten? → melden

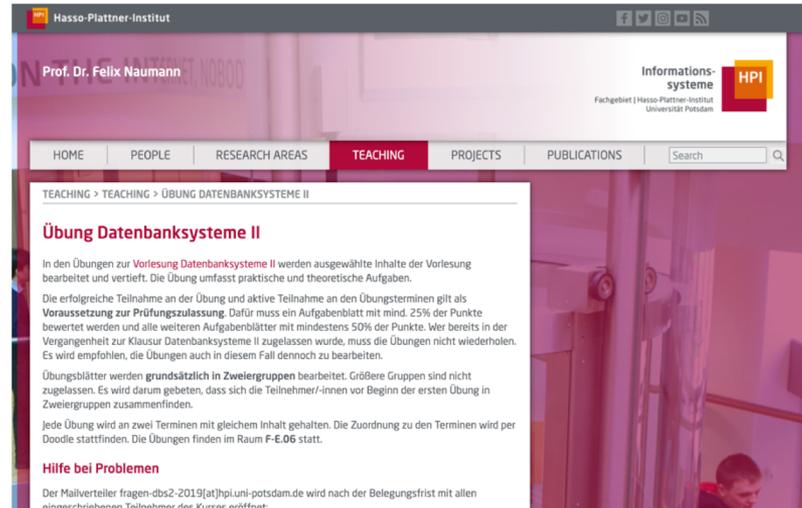
**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Organisatorisches: Übung Datenbanksysteme II

## Homepage

- Aktuelle Informationen
- Übungstermine
- Übungsblätter
- Folien



The screenshot shows the HPI website page for 'Übung Datenbanksysteme II'. The header includes the HPI logo, the name 'Prof. Dr. Felix Naumann', and the department 'Informations-systeme'. The navigation menu has 'TEACHING' selected. The main content area is titled 'Übung Datenbanksysteme II' and contains the following text:

In den Übungen zur Vorlesung **Datenbanksysteme II** werden ausgewählte Inhalte der Vorlesung bearbeitet und vertieft. Die Übung umfasst praktische und theoretische Aufgaben.

Die erfolgreiche Teilnahme an der Übung und aktive Teilnahme an den Übungsterminen gilt als **Voraussetzung zur Prüfungszulassung**. Dafür muss ein Aufgabenblatt mit mind. 25% der Punkte bewertet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte. Wer bereits in der Vergangenheit zur Klausur Datenbanksysteme II zugelassen wurde, muss die Übungen nicht wiederholen. Es wird empfohlen, die Übungen auch in diesem Fall dennoch zu bearbeiten.

Übungsblätter werden **grundsätzlich in Zweiergruppen** bearbeitet. Größere Gruppen sind nicht zugelassen. Es wird darum gebeten, dass sich die Teilnehmer/-innen vor Beginn der ersten Übung in Zweiergruppen zusammenfinden.

Jede Übung wird an zwei Terminen mit gleichem Inhalt gehalten. Die Zuordnung zu den Terminen wird per Doodle stattfinden. Die Übungen finden im Raum **F-E.06** statt.

**Hilfe bei Problemen**

Der Mailverteiler fragen-dbs2-2019[at]hpi.uni-potsdam.de wird nach der Belegungsfrist mit allen eingeschriebenen Teilnehmer des Kurses eröffnet.

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

<https://hpi.de/naumann/teaching/teaching/ws-1920/uebung-datenbanksysteme-ii.html>

# Organisatorisches: Übung Datenbanksysteme II

---

## Abgabesystem

- Abgabe immer in Zweiergruppen
- Abgabe und Korrektur nur noch digital
- URL: <http://www.dcl.hpi.uni-potsdam.de/submit/>
- Alle dort einmal für den Kurs DBS II eintragen
- Anschließend Lösungen dort hochladen

Besorgt euch Schlüssel :)

Übungsthemen  $\neq$  Hausaufgabenthemen

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

Lösungen immer begründen!

- Seid präzise (Niemand liest gerne lange Texte :) )
- Im Zweifel 1-2 Sätze mehr
- Wenn ihr Annahmen trifft, schreibt diese klar hin und argumentiert knapp warum diese sinnvoll sind

Rechenaufgaben

- In der Klausur habt ihr keine Rechenhilfe  
→ Kopfrechnen (dementsprechend werden halbwegs vernünftige Zahlen herauskommen)
- In den Übungsaufgaben ist das nicht immer der Fall

Rechentipps

- Nutzt Potenzen und kürzt Brüche:  
512 / 64 lässt sich in Exponentialform viel einfacher ausrechnen
- „Komische Zahlen“ ruhig als Variablen benutzen

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

**5**

# Organisatorisches: Übung Datenbanksysteme II

---

Fragen?

- Zum Organisatorischen?
- Zum Übungszettel?

Gebt kontinuierliches Feedback

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Erinnerung: Dezimal- vs. Binär-Einheiten

Dezimalpräfixe		Unterschied (gerundet)	Binärpräfixe	
Name (Symbol)	Bedeutung <sup>[G 1]</sup>		IEC-Name (IEC-Symbol)	Bedeutung
Kilobyte (kB) <sup>[G 2]</sup>	10 <sup>3</sup> Byte = 1.000 Byte	2,40 %	Kibibyte (KiB) <sup>[G 3]</sup>	2 <sup>10</sup> Byte = 1.024 Byte
Megabyte (MB)	10 <sup>6</sup> Byte = 1.000.000 Byte	4,86 %	Mebibyte (MiB)	2 <sup>20</sup> Byte = 1.048.576 Byte
Gigabyte (GB)	10 <sup>9</sup> Byte = 1.000.000.000 Byte	7,37 %	Gibibyte (GiB)	2 <sup>30</sup> Byte = 1.073.741.824 Byte
Terabyte (TB)	10 <sup>12</sup> Byte = 1.000.000.000.000 Byte	9,95 %	Tebibyte (TiB)	2 <sup>40</sup> Byte = 1.099.511.627.776 Byte
Petabyte (PB)	10 <sup>15</sup> Byte = 1.000.000.000.000.000 Byte	12,6 %	Pebibyte (PiB)	2 <sup>50</sup> Byte = 1.125.899.906.842.624 Byte
Exabyte (EB)	10 <sup>18</sup> Byte = 1.000.000.000.000.000.000 Byte	15,3 %	Exbibyte (EiB)	2 <sup>60</sup> Byte = 1.152.921.504.606.846.976 Byte
Zettabyte (ZB)	10 <sup>21</sup> Byte = 1.000.000.000.000.000.000.000 Byte	18,1 %	Zebibyte (ZiB)	2 <sup>70</sup> Byte = 1.180.591.620.717.411.303.424 Byte
Yottabyte (YB)	10 <sup>24</sup> Byte = 1.000.000.000.000.000.000.000.000 Byte	20,9 %	Yobibyte (YiB)	2 <sup>80</sup> Byte = 1.208.925.819.614.629.174.706.176 Byte

1. ↑ SI-Präfixe sind nur für SI-Einheiten standardisiert; Byte ist keine SI-Einheit  
2. ↑ wird gelegentlich mit „kB“ abgekürzt  
3. ↑ wird gelegentlich mit „KB“ abgekürzt, um den Unterschied zu „kB“ zu kennzeichnen (nicht standardisiert)

Für uns alles das gleiche!  
**Nur Zweierpotenzen!**

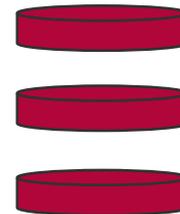
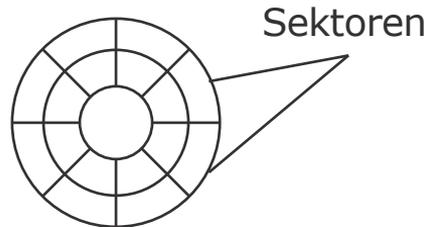
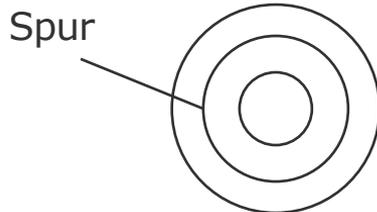
**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Übersicht: Begriffe

- **Sektor:** Abschnitt einer Spur (fester Länge)
- **Spur:** Ein Kreis auf einer Oberfläche
- **Platte und Oberfläche:** selbsterklärend
- **Zylinder:** alle Spuren eines bestimmten Radius
  - Über die gesamte Höhe
- **Block:** besteht aus 1 oder mehr Sektoren (use-case abhängig)

Auswendig lernen! :)



**Tobias Bleifuß**  
**Übung DBS II**

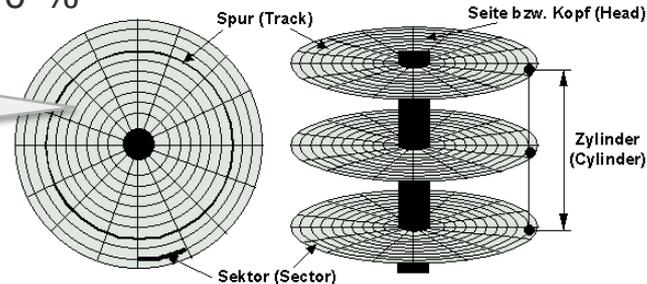
Physische  
Speicherstrukturen

# Übersicht: Physische Speicherstrukturen

Sektorengröße:	512 Byte
Sektoren pro Spur:	Ø 64
Spuren pro Oberfläche:	2.048
Oberflächen pro Platte:	2
Anzahl Platten:	5
Rotationsgeschwindigkeit:	5.000 U/min
Lesekopfbewegung über n Spuren:	$(1 + 0,002 \cdot n)$ ms
Blockgröße:	1.024 Byte
Lücken pro Spur:	10 %

0,5 ms Start  
 $0,002 \cdot n$  ms Bewegung  
0,5 ms Stop

Aufteilung der Sektoren  
auf den Spuren muss  
nicht tortenförmig sein!  
→ evtl. außen mehr Sektoren



**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 1: Speicherkapazität berechnen

Sektorengröße:	512 Byte
Sektoren pro Spur:	Ø 64
Spuren pro Oberfläche:	2.048
Oberflächen pro Platte:	2
Anzahl Platten:	5

**Gesucht:** Kapazität einer Spur

$$K_{\text{Spur}} = \text{Sektorgröße} \cdot \text{Sektoren pro Spur} = 2^9 B \cdot 2^6 = 2^{15} B$$

**Gesucht:** Kapazität einer Oberfläche

$$K_{\text{Oberfläche}} = K_{\text{Spur}} \cdot \text{Spuren pro Oberfläche} = 2^{15} B \cdot 2^{11} = 2^{26} B$$

**Gesucht:** Kapazität der Festplatte

$$K_{\text{Platte}} = K_{\text{Oberfläche}} \cdot \text{Anzahl Oberflächen} = 2^{26} B \cdot 2 \cdot 5 = 10 \cdot 2^{26} B$$

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 1: Speicherkapazität berechnen

Sektorengroße:	512 Byte
Sektoren pro Spur:	Ø 64
Spuren pro Oberfläche:	2.048
Oberflächen pro Platte:	2
Anzahl Platten:	5

**Gesucht:** Anzahl Zylinder auf dieser Festplatte

1.024	2.048	640	10.240	Weiß nicht
	= Anzahl Spuren			

# Aufgabe 1: Speicherkapazität berechnen

Sektorengröße:	512 Byte
Sektoren pro Spur:	Ø 64
Spuren pro Oberfläche:	2.048
Oberflächen pro Platte:	2
Anzahl Platten:	5

**Gesucht:** Sind die folgenden Blockgrößen zulässig?

**256 Byte**

1.024 Byte

51.200 Byte

Ja

Nein

Weiß nicht

Blöcke  
können nicht  
kleiner als 1  
Sektor sein

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 1: Speicherkapazität berechnen

Sektorengroße:	512 Byte
Sektoren pro Spur:	Ø 64
Spuren pro Oberfläche:	2.048
Oberflächen pro Platte:	2
Anzahl Platten:	5

**Gesucht:** Sind die folgenden Blockgrößen zulässig?

256 Byte

**1.024 Byte**

51.200 Byte

**Ja**

**Nein**

**Weiß nicht**

Block  
umfasst  
genau 2  
Sektoren

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 1: Speicherkapazität berechnen

Sektorengröße:	512 Byte
Sektoren pro Spur:	Ø 64
Spuren pro Oberfläche:	2.048
Oberflächen pro Platte:	2
Anzahl Platten:	5

**Gesucht:** Sind die folgenden Blockgrößen zulässig?

256 Byte

1.024 Byte

**51.200 Byte**

Ja

Nein

Weiß nicht

Pro Spur  
gibt es nur  
 $2^{15} B =$   
32.768 Byte

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 2: Latenzzeiten Begriffe

- **Kommunikationszeit:**
  - Zeit, die der Lesekopf braucht um die Informationen über den Bus zu schicken (vernachlässigbar klein)
- **Suchzeit** (seek time)
  - Zeit, die der Lesekopf braucht um die richtige Spur zu erreichen
- **Rotationszeit** (rotational latency)
  - Zeit, die der Lesekopf braucht bis der richtige Teil der Spur unter ihn rotiert ist.
- **Transferzeit** (transfer time)
  - Zeit, die der Lesekopf braucht um alle Daten zu lesen
- **Latenzzeit** (latency)
  - Summe aus allen

Auswendig lernen! :)

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 2: Latenzzeiten berechnen

Sektorengröße:	512 Byte
Sektoren pro Spur:	Ø 64
Spuren pro Oberfläche:	2.048
Rotationsgeschwindigkeit:	5.000 U/min
Lesekopfbewegung über n Spuren:	$(1 + 0,002 \cdot n)$ ms
Blockgröße:	1.024 Byte
Lücken pro Spur:	10 %

**Gesucht:** Minimale, maximale und durchschnittliche Latenzzeit  
zum Lesen eines Blocks

Latenzzeit = Kommunikation + Such + Rotation + Transfer

vernachlässigbar

für alle drei Fälle gleich

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 2: Latenzzeiten berechnen

Sektorgröße:	512 Byte	Lesekopfbewegung über n Spuren:	
Sektoren pro Spur:	Ø 64	(1 + 0,002 · n) ms	
Spuren pro Oberfläche:	2048	Blockgröße:	1024 Byte
Rotationsgeschwindigkeit:	5000 U/min	Lücken pro Spur:	10 %

**Gesucht:** Minimale, maximale und durchschnittliche Latenzzeit zum Lesen eines Blocks

Latenzzeit = Kommunikation + Seek + Rotation + Transfer

Umdrehungszeit =  $1 \text{ U} / \text{Rotationsgeschw.} = 1 / 5000 \text{ min} = 12 \text{ ms}$

$T_{\text{sektor}} = 12 \text{ms} * 0.9 / 64 = 0,16875 \text{ms}$

$T_{\text{Lücke}} = 12 \text{ms} * 0.1 / 64 = 0,01875 \text{ms}$

Strecke = 2 Sektoren + 1 Lücke (2. Lücke ist ja egal)

$L_{\text{Transfer}} = 0,16875 \text{ms} * 2 + 0,01875 \text{ms} = 0.35625 \text{ms}$

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 2: Latenzzeiten berechnen

Sektorgröße:	512 Byte	Lesekopfbewegung über n Spuren:	
Sektoren pro Spur:	Ø 64	(1 + 0,002 · n) ms	
Spuren pro Oberfläche:	2048	Blockgröße:	1024 Byte
Rotationsgeschwindigkeit:	5000 U/min	Lücken pro Spur:	10 %

**Gesucht:** Minimale, maximale und durchschnittliche Latenzzeit zum Lesen eines Blocks

Latenzzeit = Kommunikation + Seek + Rotation + Transfer

$$\begin{aligned}L_{\text{Transfer\_min}} &= 0,35625 \text{ ms} && (= L_{\text{Transfer}}) \\L_{\text{Seek\_min}} &= 0 \text{ ms} && (\text{Lesekopf auf passender Spur}) \\L_{\text{Rotation\_min}} &= 0 \text{ ms} && (\text{Lesekopf genau vor dem Block}) \\L_{\text{min}} &= 0,35625 \text{ ms} + 0 \text{ ms} + 0 \text{ ms} = 0,35625 \text{ ms}\end{aligned}$$

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 2: Latenzzeiten berechnen

Sektorgröße:	512 Byte	Lesekopfbewegung über n Spuren:	
Sektoren pro Spur:	Ø 64	(1 + 0,002 · n) ms	
Spuren pro Oberfläche:	2048	Blockgröße:	1024 Byte
Rotationsgeschwindigkeit:	5000 U/min	Lücken pro Spur:	10 %

**Gesucht:** Minimale, maximale und durchschnittliche Latenzzeit zum Lesen eines Blocks

Latenzzeit = Kommunikation + Seek + Rotation + Transfer

$$\begin{aligned}L_{\text{Transfer\_max}} &= 0,35625 \text{ ms} \quad (= L_{\text{Transfer}}) \\L_{\text{Seek\_max}} &= (1 + 0,002 \cdot 2047) \text{ ms} = 5,094 \text{ ms} \quad (\text{über alle Spuren}) \\L_{\text{Rotation\_max}} &= 12 \text{ ms} \quad (\text{volle Rotation}) \\L_{\text{max}} &= 0,35625 \text{ ms} + 5,094 \text{ ms} + 12 \text{ ms} = 17,45025 \text{ ms}\end{aligned}$$

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 2: Latenzzeiten berechnen

Sektorgröße:	512 Byte	Lesekopfbewegung über n Spuren:	
Sektoren pro Spur:	Ø 64	(1 + 0,002 · n) ms	
Spuren pro Oberfläche:	2048	Blockgröße:	1024 Byte
Rotationsgeschwindigkeit:	5000 U/min	Lücken pro Spur:	10 %

**Gesucht:** Minimale, maximale und durchschnittliche Latenzzeit zum Lesen eines Blocks

Latenzzeit = Kommunikation + Seek + Rotation + Transfer

$$\begin{aligned}L_{\text{Transfer\_avg}} &= 0,35625 \text{ ms} \quad (= L_{\text{Transfer}}) \\L_{\text{Seek\_avg}} &= (1 + 0,002 \cdot 2048 / 3) \text{ ms} \approx 2,355 \text{ ms} \quad (\text{avg. Distanz}) \\L_{\text{Rotation\_avg}} &= 12 \text{ ms} / 2 = 6 \text{ ms} \quad (\text{halbe Rotation}) \\L_{\text{avg}} &= 0,35625 \text{ ms} + 2,355 \text{ ms} + 6 \text{ ms} = 8,71125 \text{ ms}\end{aligned}$$

“1/3 der Spuren”  
ist eine gute An-  
näherung

$\neq L_{\text{Seek\_max}} / 2$

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 3: Daten speichern

Sektorgröße:	512 Byte	Oberflächen pro Platte:	2
Sektoren pro Spur:	Ø 64	Anzahl Platten:	5
Spuren pro Oberfläche:	2048	Blockgröße:	1024 Byte

Datei mit 100.000 Tupeln der Größe 100 Byte

Es gibt keine Tupel, die auf mehrere Blöcke aufgeteilt sind

a) **Gesucht:** Anzahl Tupel pro Block

<b>100</b>	<b>11</b>	<b>10</b>	<b>20</b>	<b>Weiß nicht</b>
		$\left\lfloor \frac{\text{Blockgröße}}{\text{Tupelgröße}} \right\rfloor$ $= \left\lfloor \frac{1024 \text{ Byte}}{100 \text{ Byte}} \right\rfloor$ $= 10$		

**Tobias Bleifuß**  
**Übung DBS II**  
Physische  
Speicherstrukturen

# Aufgabe 3: Daten speichern

Sektorgröße:	512 Byte	Oberflächen pro Platte:	2
Sektoren pro Spur:	Ø 64	Anzahl Platten:	5
Spuren pro Oberfläche:	2048	Blockgröße:	1024 Byte

Datei mit 100.000 Tupeln der Größe 100 Byte

Es gibt keine Tupel, die auf mehrere Blöcke aufgeteilt sind

b) **Gesucht:** Anzahl Blöcke für vollständige Datei

<b>100</b>	<b>1000</b>	<b>10</b>	<b>10.000</b>	<b>Weiß nicht</b>
			$\left\lceil \frac{\#Tupel}{Tupel \text{ pro Block}} \right\rceil$ $= \left\lceil \frac{100.000}{100} \right\rceil$ $= 10.000$	

**Tobias Bleifuß**  
**Übung DBS II**  
Physische  
Speicherstrukturen

# Aufgabe 3: Daten speichern

Sektorgröße:	512 Byte	Oberflächen pro Platte:	2
Sektoren pro Spur:	Ø 64	Anzahl Platten:	5
Spuren pro Oberfläche:	2048	Blockgröße:	1024 Byte

Datei mit 100.000 Tupeln der Größe 100 Byte

Es gibt keine Tupel, die auf mehrere Blöcke aufgeteilt sind

c) **Gesucht:** Maximale Anzahl Tupel auf der gesamten Festplatte

$$\begin{aligned} \text{Blöcke pro Spur} &= \text{Sektoren pro Spur} / \text{Sektoren pro Block} \\ &= 2^6 B / (2^{10} B / 2^9 B) = 2^5 \end{aligned}$$

$$\begin{aligned} \text{Tupel pro Festplatte} &= \text{Tupel pro Block} \cdot \text{Blöcke pro Spur} \cdot \text{Spuren pro Oberfläche} \\ &\quad \cdot \text{Oberflächen pro Platte} \cdot \text{Anzahl Platten} \\ &= 10 \cdot 2^5 \cdot 2^{11} \cdot 2 \cdot 5 = 100 \cdot 2^{16} \end{aligned}$$

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

**23**

# Aufgabe 4: Daten lesen

- Anzahl Zylinder: 8192
- Seekzeit für n Zylinder:  $(1 + 0,002 \cdot n)$  ms
- Rotationszeit:  $\emptyset$  6,5 ms
- Transferzeit:  $\emptyset$  0,5 ms
- Initiale Lesekopfposition: 4000

- Anfragen: 

Eintreffen der Anfrage	0 ms	3 ms	11 ms	19 ms
Angefragter Zylinder	6500	2000	8000	3500

- a) **Gesucht:** Bearbeitung der Anfragen mit First-Come, First-Served
- b) **Gesucht:** Bearbeitung der Anfragen mit dem Elevator Algorithmus

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Daten Lesen

## Wiederholung: Elevator Algorithm

- Lesekopf streicht über die Oberfläche
  - Hält an Zylinder an wenn dort IO-Anfrage benötigt wird
  - Dreht um falls in der momentanen Richtung keine IO-Anfrage mehr zu bearbeiten ist

### Disk Scheduling



Idee: Disk-Controller entscheidet, welche Zugriffsanweisungen zuerst ausgeführt werden.

- Nützlich bei vielen kleinen Prozessen, je auf wenigen Blöcken
  - OLTP
  - Ziel: Erhöhung des Durchsatzes
- **Elevator Algorithmus**
  - Fahrstuhl fährt in Gebäude hoch und runter
    - Hält an Stockwerken an, wenn jemand ein- oder aussteigen will.
    - Dreht um, falls weiter oben/unten keiner mehr wartet.
  - Diskkopf streicht über Oberfläche einwärts und auswärts
    - Hält an Zylindern an, wenn es eine (oder mehrere) Zugriffsanweisung(en) gibt.
    - Dreht um, falls in jeweiliger Richtung keine Anweisungen mehr ausstehen.

Felix Naumann  
Datenbanksysteme II  
Winter 2019/20

55

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

25

# Aufgabe 4: Daten lesen

Eintreffen der Anfrage	0 ms	3 ms	11 ms	19 ms
Angefragter Zylinder	6500	2000	8000	3500

Anzahl Zylinder: 8192  
Seekzeit für n Zylinder:  $(1 + 0,002 \cdot n)$  ms  
Rotationszeit:  $\emptyset$  6,5 ms  
Transferzeit:  $\emptyset$  0,5 ms  
Initiale Lesekopfposition: 4000

## Anfangsbeispiel

Start Time	Lesekopf-Position	Request Queue	End Time	Direction
0ms	4000	6500		

# Aufgabe 4: Daten lesen

Eintreffen der Anfrage	0 ms	3 ms	11 ms	19 ms
Angefragter Zylinder	6500	2000	8000	3500

Anzahl Zylinder: 8192  
 Seekzeit für n Zylinder:  $(1 + 0,002 \cdot n)$  ms  
 Rotationszeit:  $\emptyset$  6,5 ms  
 Transferzeit:  $\emptyset$  0,5 ms  
 Initiale Lesekopfposition: 4000

## Anfangsbeispiel

Start Time	Lesekopf-Position	Request Queue	End Time	Direction
0ms	4000	6500	$(1+0,002 \cdot 2500)+6,5+0,5=13ms$	→
13ms				

# Aufgabe 4: Daten lesen

Eintreffen der Anfrage	0 ms	3 ms	11 ms	19 ms
Angefragter Zylinder	6500	2000	8000	3500

Anzahl Zylinder: 8192  
Seekzeit für n Zylinder:  $(1 + 0,002 \cdot n)$  ms  
Rotationszeit:  $\emptyset$  6,5 ms  
Transferzeit:  $\emptyset$  0,5 ms  
Initiale Lesekopfposition: 4000

**Gesucht:** Zeit für das Lesen aller Blöcke nach **first-come, first-served**

<b>52ms</b>	<b>57ms</b>	<b>62ms</b>	<b>67ms</b>	<b>Weiß nicht</b>
<input type="text"/>				

**Tobias Bleifuß**  
**Übung DBS II**  
Physische  
Speicherstrukturen

# Aufgabe 4: Daten lesen

Eintreffen der Anfrage	0 ms	3 ms	11 ms	19 ms
Angefragter Zylinder	6500	2000	8000	3500

Anzahl Zylinder: 8192  
 Seekzeit für n Zylinder:  $(1 + 0,002 \cdot n)$  ms  
 Rotationszeit:  $\emptyset$  6,5 ms  
 Transferzeit:  $\emptyset$  0,5 ms  
 Initiale Lesekopfposition: 4000

## First-Come, First-Served

Start Time	Lesekopf-Position	Request Queue	End Time	Direction
0ms	4000	6500	$(1+0,002 \cdot 2500)+6,5+0,5=13\text{ms}$	→
13ms	6500	2000 8000	$(1+0,002 \cdot 4500)+6,5+0,5=17\text{ms}$	←
30ms	2000	8000 3500	$(1+0,002 \cdot 6000)+6,5+0,5=20\text{ms}$	→
50ms	8000	3500	$(1+0,002 \cdot 4500)+6,5+0,5=17\text{ms}$	←
<b>67ms</b>				

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
 Speicherstrukturen

# Aufgabe 4: Daten lesen

Eintreffen der Anfrage	0 ms	3 ms	11 ms	19 ms
Angefragter Zylinder	6500	2000	8000	3500

Anzahl Zylinder: 8192  
Seekzeit für n Zylinder:  $(1 + 0,002 \cdot n)$  ms  
Rotationszeit:  $\emptyset$  6,5 ms  
Transferzeit:  $\emptyset$  0,5 ms  
Initiale Lesekopfposition: 4000

**Gesucht:** Zeit für das Lesen aller Blöcke mittels **Elevator-Algorithmus**

<b>52ms</b>	<b>57ms</b>	<b>62ms</b>	<b>67ms</b>	<b>Weiß nicht</b>
				

**Tobias Bleifuß**  
**Übung DBS II**  
Physische  
Speicherstrukturen

# Aufgabe 4: Daten lesen

Eintreffen der Anfrage	0 ms	3 ms	11 ms	19 ms
Angefragter Zylinder	6500	2000	8000	3500

Anzahl Zylinder: 8192  
 Seekzeit für n Zylinder:  $(1 + 0,002 \cdot n)$  ms  
 Rotationszeit:  $\emptyset$  6,5 ms  
 Transferzeit:  $\emptyset$  0,5 ms  
 Initiale Lesekopfposition: 4000

## Elevator-Algorithmus

Start Time	Lesekopf-Position	Request Queue	End Time	Direction
0ms	4000	6500	$(1+0,002 \cdot 2500)+6,5+0,5=13\text{ms}$	→
13ms	6500	8000 2000	$(1+0,002 \cdot 1500)+6,5+0,5=11\text{ms}$	→
24ms	8000	3500 2000	$(1+0,002 \cdot 4500)+6,5+0,5=17\text{ms}$	←
41ms	3500	2000	$(1+0,002 \cdot 1500)+6,5+0,5=11\text{ms}$	←
<b>52ms</b>				

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
 Speicherstrukturen

**31**

# Aufgabe 5:

## Daten auf verschiedenen Spuren

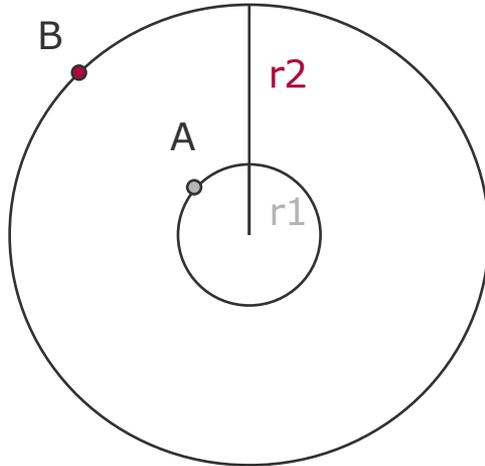
---

- Rotations- und Lesekopfgeschwindigkeit sind konstant, d.h.
  - Rotationslatenz unverändert
  - sequentielle Datentransferrate ist auf äußeren Spuren größer, falls dort mehr Sektoren pro Spur angelegt wurden
  - Was ist mit der Seektime?
    - ◇ Geschwindigkeit des Lesekopfes ist gleich...
    - ◇ ... aber die durchschnittliche Anzahl an zu überquerenden Spuren ist in der Mitte am geringsten

# Aufgabe 5:

## Daten auf verschiedenen Spuren

Warum ist Lesen/Schreiben auf der äußeren Spur schneller?



- Beide Punkte legen eine Umdrehung in der gleichen Zeit zurück (Gleiche Winkelgeschwindigkeit)
- Zurückgelegte Strecke = Umfang Kreis =  $2 \cdot \pi \cdot r$
- $r1 < r2 \rightarrow$  B bewegt sich mit größerer Geschwindigkeit als A (Bahngeschwindigkeit)
- $\rightarrow$  Sektoren die auf Spur B liegen werden schneller gelesen

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 5:

## Daten auf verschiedenen Spuren

---

Wo sollte man daher die folgenden Dateien für die genannten Zugriffe positionieren (innen, mittig, außen)?

- seltene, sequentielle Scans ...
  - einer großen Datei
  - einer kleinen Datei
- häufiger, random Zugriff auf ...
  - eine kleine Datei
  - eine große Datei per Index

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 5:

## Daten auf verschiedenen Spuren

Wo sollte man daher die folgenden Dateien für die genannten Zugriffe positionieren (innen, mittig, außen)?

- seltene, sequentielle Scans ...
  - einer großen Datei:

innen	mittig	außen	Weiß nicht
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 5:

## Daten auf verschiedenen Spuren

Wo sollte man daher die folgenden Dateien für die genannten Zugriffe positionieren (innen, mittig, außen)?

- seltene, sequentielle Scans ...
  - einer kleinen Datei:

innen	mittig	außen	Weiß nicht
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 5:

## Daten auf verschiedenen Spuren

Wo sollte man daher die folgenden Dateien für die genannten Zugriffe positionieren (innen, mittig, außen)?

■ seltene, sequentielle Scans ...

□ einer großen Datei:

- Kosten dominiert durch sequentiellen Datentransfer
- Sequentieller Datentransfer ist außen am schnellsten

□ einer kleinen Datei:

- Kosten dominiert durch initialen Seek und Rotation (Lesen einer kleinen Datei ist effektiv Random I/O)
- Innen wird nichts optimiert, aber das Lagern kleiner, selten zugriffener Dateien tut hier am wenigsten weh

# Aufgabe 5:

## Daten auf verschiedenen Spuren

Wo sollte man daher die folgenden Dateien für die genannten Zugriffe positionieren (innen, mittig, außen)?

- häufiger, random Zugriff auf ...
  - eine kleine Datei:

innen	mittig	außen	Weiß nicht
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 5:

## Daten auf verschiedenen Spuren

Wo sollte man daher die folgenden Dateien für die genannten Zugriffe positionieren (innen, mittig, außen)?

- häufiger, random Zugriff auf ...
  - eine große Datei per Index:

innen	mittig	außen	Weiß nicht
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Aufgabe 5:

## Daten auf verschiedenen Spuren

Wo sollte man daher die folgenden Dateien für die genannten Zugriffe positionieren (innen, mittig, außen)?

■ häufiger, random Zugriff auf ...

□ eine kleine Datei:

- Kosten dominiert durch Seek und Rotation
- Seek wird in der Mitte minimiert  
(Wegen häufigem Zugriff ist die Optimierung der Seektime hier am wichtigsten)

Annahme: Index Steht bei Datei

□ eine große Datei per Index:

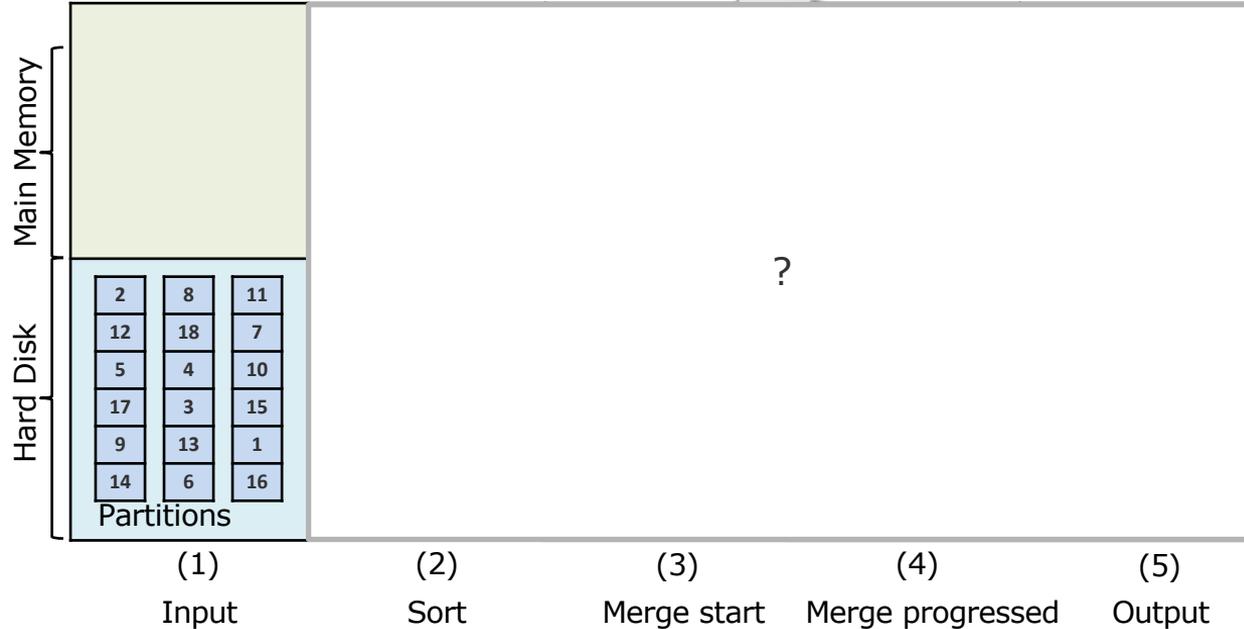
- Kosten dominiert durch häufigen Seek zwischen Datei und Index
- Seek wird durch nahes Zusammenlegen minimiert
- Beim Platzieren innen sparen wir den jeweils wertvollen äußeren und mittleren Platz

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

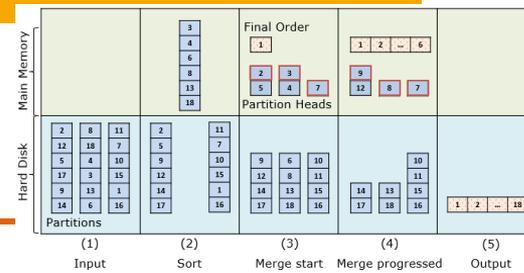
Prefetching! Vorteil?

Algorithmus:



## Aufgabe 6:

# Daten sortieren: TPMMS



## Gedankenspiel: Three-Phase-Multiway-Merge-Sort

1. Warum könnte er notwendig sein?

Es passen nicht alle Köpfe vorsortierter Teillisten in den RAM.

2. Wie könnte ein Three-Phase-Multiway-Merge-Sort funktionieren?

P1: Sort; P2: (Pre-)Merge jeweils  $n$  Teillisten; P3: Merge alle Listen

Um wie viel steigen die Lese- und Schreibkosten?

1 x alle Tupel lesen + 1 x alle Tupel schreiben

3. Die Partitionen müssen nicht unbedingt Hauptspeicher-groß sein. Welche Vorteile könnte eine Partitionsgröße  $\frac{\text{Hauptspeichergröße}}{2}$  haben?

Sortieren im RAM müsste nicht in-place sein und ist so  $O(n \cdot \log(n))$

## Aufgabe 1: Speichermenge und Lesezeit

Gegeben sei eine Festplatte mit folgenden Merkmalen:

- Es gibt fünf Platten mit  $9\text{cm}$  Durchmesser, die jeweils zwei Plattenoberflächen besitzen.
- Jede Plattenoberfläche trägt 20.000 Spuren.
- Jede Spur besteht durchschnittlich aus 1.000 Sektoren der Größe 512 Byte.
- 5 % jeder Spur werden von Lücken überdeckt.
- Die Platten rotieren mit einer Geschwindigkeit von 10.000 U/min.
- Die Bewegung des Lesekopfes über  $n$  Spuren benötigt  $(1 + 0,002n)$  ms.
- Ein Block umfasst 5 Sektoren.

- a) Was ist die Gesamtkapazität der Platte in GB? **1 P**
- b) Wie viele Blöcke liegen auf einem Zylinder? **1 P**
- c) Was ist die durchschnittliche Latenzzeit für einen Block? **2 P**
- d) Auf dieser Festplatte sei eine Datei gespeichert, die eine komplette Spur belegt. Wie lange dauert durchschnittlich das sequentielle Einlesen der gesamten Datei, wenn das Einlesen einer Spur nur an einer bestimmten Position auf der Spur beginnen darf? **2 P**

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 2: Two-Phase Multiway Merge Sort

Gegeben sei die Festplatte aus Aufgabe 1 und ein Hauptspeicher von 200 MiB. Wir betrachten in dieser Aufgabe den naiven Two-Phase Multiway Merge Sort Algorithmus, das heißt: In Phase 2 wird jeweils nur der erste Block jeder Teilliste in den Hauptspeicher geladen und jeder Block wird einzeln auf die finale Ausgabeliste geschrieben.

- a) Wie lange dauert das Lesen und Schreiben von Tupeln insgesamt, wenn 20 Millionen Tupel sortiert werden müssen. Ein Block umfasst 100 Tupel. Zur Optimierung der Zugriffsgeschwindigkeit sind die Daten zuvor optimal über die Spuren einer Oberfläche verteilt worden (d.h. die Daten befinden sich sequentiell hintereinander).

*HINWEIS: Nutzen Sie die Ergebnisse aus Aufgabe 1!*

**9 P**

- b) Was ist die Mindestgröße des Hauptspeichers (in Anzahl von Blöcken), um die 20 Millionen Tupel aus Aufgabe a) per Two-Phase Multiway Merge Sort zu sortieren? Begründen Sie Ihre Antwort!

**3 P**

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 3: Datensätze fester vs. variabler Länge

Gegeben seien die folgenden Anforderungen für eine Relation einer Patientendatenbank: Es gibt 10 Attribute, die immer gefüllt werden (z. B. Name, Geburtsdatum) und weitere 20 optionale Attribute, die nicht für jeden Patienten bekannt oder relevant sind (z. B. aktuelle Blutzuckerwerte). Jedes der optionalen Attribute wird mit der Wahrscheinlichkeit  $p$  gefüllt.

Die Werte der erforderlichen Attribute werden in Feldern der definierten Länge von 64 Byte gespeichert, die Werte der optionalen Attribute in Feldern der festen Länge von 8 Byte. Desweiteren nehmen wir an, dass die Tupel lediglich die Daten und keinerlei Header-Informationen speichern.

- a) Was ist die erwartete Größe eines Tupels (in Abhängigkeit von  $p$ ) unter Verwendung von
- Datensätzen fester Länge? **1 P**
  - Datensätzen variabler Länge? Die optionalen Attribute sollen als getaggte Felder gespeichert werden. Das Tag jedes dieser Attribute umfasst 2 Byte und die Sequenz der Felder wird mit einem gesonderten Tag der Größe 2 Byte abgeschlossen. **1 P**
- b) Für welche Wahrscheinlichkeiten  $p$  sind Datensätze fester Länge zu bevorzugen? **2 P**

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 4: Speichern von Relationen

Gegeben seien die folgenden beiden Relationen Hersteller und Produkte, wobei Produkte per Fremdschlüssel einem Hersteller zugeordnet sind. Alle Tupel werden als Datensätze fester Länge gespeichert.

Die Hersteller-Relation umfasst 10.000 Tupel. Jeder Datensatz ist 500 Byte lang. Die Produkte-Relation umfasst 40.000 Tupel. Jeder Produkte-Datensatz hat die Größe 200 Byte.

Die Datensätze sollen in Blöcken der Größe 4.096 Byte gespeichert werden, wobei jeder Block 64 Byte für Header-Informationen reserviert. Es gibt keine Datensätze, die auf mehrere Blöcke aufgeteilt sind.

- a) Wie viele Blöcke werden benötigt, um beide Relationen zu speichern, wenn jede Relation am Stück (in getrennte Blockgruppen) gespeichert wird? **4 P**
- b) Wie viele Blöcke werden benötigt, um beide Relationen zu speichern, wenn jeder Hersteller und seine Produkte im selben Block gespeichert werden? Nimm an, dass jeder Hersteller 4 Produkte herstellt. **4 P**
- c) Der Workload auf der Datenbank umfasst zwei Typen von Anfragen:
  - 1) Lesen aller Hersteller-Daten
  - 2) Join der Relationen Hersteller und Produkte

Welche der beiden obigen Speichervarianten (Aufgabenteil a und b) sollten für welche Anfrageart bevorzugt werden? Und warum? (Ein Satz als Antwort genügt.) **4 P**

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

## Aufgabe 5: Verschiedene Lesegeschwindigkeiten

Gegeben sei eine Festplatte, bei der der Radius der äußersten Spur doppelt so groß ist wie der Radius der innersten Spur:  $r_A = 2 \cdot r_I$ . Wir nehmen an, dass die Größe eines Sektors konstant ist, d.h. ein Sektor auf einer äußeren Spur ist genauso lang und fasst genauso viel Speicher wie ein Sektor auf einer inneren Spur, das gleiche gilt für die Länge der Lücken. Wie viel kleiner ist die Transferzeit eines Sektors auf der äußersten Spur im Verhältnis zur Transferzeit auf der innersten Spur?

**2 P**

**Tobias Bleifuß**  
**Übung DBS II**

Physische  
Speicherstrukturen

# Laserpointer

Sollen wir in den nächsten Übungsterminen wieder Laserpointer verwenden?

Ja	Nein	Mir egal
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>