Part I

Organizational Matters
Part I

Organizational Matters

► Modul: IN2003

► Name: “Efficient Algorithms and Data Structures”
  “Effiziente Algorithmen und Datenstrukturen”

► ECTS: 8 Credit points

► Lectures:
  ► 4 SWS
    Mon 10:00–12:00 (Room Interim2)
    Fri 10:00–12:00 (Room Interim2)

► Webpage: http://www14.in.tum.de/lehre/2017WS/ea/
Part I

Organizational Matters

- Modul: IN2003
- Name: “Efficient Algorithms and Data Structures”
  “Effiziente Algorithmen und Datenstrukturen”
- ECTS: 8 Credit points
- Lectures:
  - 4 SWS
    Mon 10:00–12:00 (Room Interim2)
    Fri 10:00–12:00 (Room Interim2)
Part I

Organizational Matters

▶ Modul: IN2003
▶ Name: “Efficient Algorithms and Data Structures”
   “Effiziente Algorithmen und Datenstrukturen”
▶ ECTS: 8 Credit points
▶ Lectures:
  ▶ 4 SWS
  Mon 10:00–12:00 (Room Interim2)
  Fri 10:00–12:00 (Room Interim2)
▶ Webpage: http://www14.in.tum.de/lehre/2017WS/ea/
Part I

Organizational Matters

- Modul: IN2003
- Name: “Efficient Algorithms and Data Structures”
  “Effiziente Algorithmen und Datenstrukturen”
- ECTS: 8 Credit points
- Lectures:
  - 4 SWS
    Mon 10:00–12:00 (Room Interim2)
    Fri 10:00–12:00 (Room Interim2)

Webpage: http://www14.in.tum.de/lehre/2017WS/ea/
Part I

Organizational Matters

- Modul: IN2003
- Name: “Efficient Algorithms and Data Structures”
  “Effiziente Algorithmen und Datenstrukturen”
- ECTS: 8 Credit points
- Lectures:
  - 4 SWS
    Mon 10:00–12:00 (Room Interim2)
    Fri 10:00–12:00 (Room Interim2)
Required knowledge:

- IN0001, IN0003
  “Introduction to Informatics 1/2”
  “Einführung in die Informatik 1/2”
- IN0007
  “Fundamentals of Algorithms and Data Structures”
  “Grundlagen: Algorithmen und Datenstrukturen” (GAD)
- IN0011
  “Basic Theoretic Informatics”
  “Einführung in die Theoretische Informatik” (THEO)
- IN0015
  “Discrete Structures”
  “Diskrete Strukturen” (DS)
- IN0018
  “Discrete Probability Theory”
  “Diskrete Wahrscheinlichkeitslehre” (DWT)
Required knowledge:

- **IN0001, IN0003**
  
  "Introduction to Informatics 1/2"
  
  "Einführung in die Informatik 1/2"

- **IN0007**
  
  "Fundamentals of Algorithms and Data Structures"
  
  "Grundlagen: Algorithmen und Datenstrukturen" (GAD)

- **IN0011**
  
  "Basic Theoretic Informatics"
  
  "Einführung in die Theoretische Informatik" (THEO)

- **IN0015**
  
  "Discrete Structures"
  
  "Diskrete Strukturen" (DS)

- **IN0018**
  
  "Discrete Probability Theory"
  
  "Diskrete Wahrscheinlichkeitstheorie" (DWT)
Required knowledge:

- IN0001, IN0003
  "Introduction to Informatics 1/2"
  "Einführung in die Informatik 1/2"

- IN0007
  "Fundamentals of Algorithms and Data Structures"
  "Grundlagen: Algorithmen und Datenstrukturen" (GAD)

- IN0011
  "Basic Theoretic Informatics"
  "Einführung in die Theoretische Informatik" (THEO)

- IN0015
  "Discrete Structures"
  "Diskrete Strukturen" (DS)

- IN0018
  "Discrete Probability Theory"
  "Diskrete Wahrscheinlichkeitstheorie" (DWT)
Required knowledge:

- IN0001, IN0003
  “Introduction to Informatics 1/2”
  “Einführung in die Informatik 1/2”

- IN0007
  “Fundamentals of Algorithms and Data Structures”
  “Grundlagen: Algorithmen und Datenstrukturen” (GAD)

- IN0011
  “Basic Theoretic Informatics”
  “Einführung in die Theoretische Informatik” (THEO)

- IN0015
  “Discrete Structures”
  “Diskrete Strukturen” (DS)

- IN0018
  “Discrete Probability Theory”
  “Diskrete Wahrscheinlichkeitstheorie” (DWT)
Required knowledge:

- IN0001, IN0003
  “Introduction to Informatics 1/2”
  “Einführung in die Informatik 1/2”

- IN0007
  “Fundamentals of Algorithms and Data Structures”
  “Grundlagen: Algorithmen und Datenstrukturen” (GAD)

- IN0011
  “Basic Theoretic Informatics”
  “Einführung in die Theoretische Informatik” (THEO)

- IN0015
  “Discrete Structures”
  “Diskrete Strukturen” (DS)

- IN0018
  “Discrete Probability Theory”
  “Diskrete Wahrscheinlichkeitstheorie” (DWT)
Required knowledge:

- IN0001, IN0003
  “Introduction to Informatics 1/2”
  “Einführung in die Informatik 1/2”

- IN0007
  “Fundamentals of Algorithms and Data Structures”
  “Grundlagen: Algorithmen und Datenstrukturen” (GAD)

- IN0011
  “Basic Theoretic Informatics”
  “Einführung in die Theoretische Informatik” (THEO)

- IN0015
  “Discrete Structures”
  “Diskrete Strukturen” (DS)

- IN0018
  “Discrete Probability Theory”
  “Diskrete Wahrscheinlichkeitstheorie” (DWT)
The Lecturer

- Harald Räcke
- Email: raecke@in.tum.de
- Room: 03.09.044
- Office hours: (by appointment)
Tutorials

A01  Monday, 12:00–14:00, 00.08.038 (Schmid)
A02  Monday, 12:00–14:00, 00.09.038 (Stotz)
A03  Monday, 14:00–16:00, 02.09.023 (Liebl)

B04  Tuesday, 10:00–12:00, 00.08.053 (Schmid)
B05  Tuesday, 12:00–14:00, 03.11.018 (Kraft)
B06  Tuesday, 14:00–16:00, 00.08.038 (Somogyi)

D07  Thursday, 10:00–12:00, 03.11.018 (Liebl)

E08  Friday, 12:00–14:00, 00.13.009 (Stotz)
E09  Friday, 14:00–16:00, 00.13.009 (Kraft)
Assignment sheets

In order to pass the module you need to pass an exam.
Assessment

Assignment Sheets:

- An assignment sheet is usually made available on Monday on the module webpage.
- Solutions have to be handed in in the following week before the lecture on Monday.
- You can hand in your solutions by putting them in the mailbox "Efficient Algorithms" on the basement floor in the MI-building.
- Solutions have to be given in English.
- Solutions will be discussed in the tutorial of the week when the sheet has been handed in, i.e., sheet may not be corrected by this time.
- You can submit solutions in groups of up to 2 people.
Assessment

Assignment Sheets:

▶ An assignment sheet is usually made available on Monday on the module webpage.

▶ Solutions have to be handed in in the following week before the lecture on Monday.

▶ You can hand in your solutions by putting them in the mailbox "Efficient Algorithms" on the basement floor in the MI-building.

▶ Solutions have to be given in English.

▶ Solutions will be discussed in the tutorial of the week when the sheet has been handed in, i.e., sheet may not be corrected by this time.

▶ You can submit solutions in groups of up to 2 people.
Assessment

Assignment Sheets:

▶ An assignment sheet is usually made available on Monday on the module webpage.

▶ Solutions have to be handed in in the following week before the lecture on Monday.

▶ You can hand in your solutions by putting them in the mailbox "Efficient Algorithms" on the basement floor in the MI-building.

▶ Solutions have to be given in English.

▶ Solutions will be discussed in the tutorial of the week when the sheet has been handed in, i.e, sheet may not be corrected by this time.

▶ You can submit solutions in groups of up to 2 people.
Assessment

Assignment Sheets:

▶ An assignment sheet is usually made available on Monday on the module webpage.

▶ Solutions have to be handed in in the following week before the lecture on Monday.

▶ You can hand in your solutions by putting them in the mailbox "Efficient Algorithms" on the basement floor in the MI-building.

▶ Solutions have to be given in English.

▶ Solutions will be discussed in the tutorial of the week when the sheet has been handed in, i.e, sheet may not be corrected by this time.

▶ You can submit solutions in groups of up to 2 people.
Assessment

Assignment Sheets:

▶ An assignment sheet is usually made available on Monday on the module webpage.

▶ Solutions have to be handed in in the following week before the lecture on Monday.

▶ You can hand in your solutions by putting them in the mailbox "Efficient Algorithms" on the basement floor in the MI-building.

▶ Solutions have to be given in English.

▶ Solutions will be discussed in the tutorial of the week when the sheet has been handed in, i.e, sheet may not be corrected by this time.

▶ You can submit solutions in groups of up to 2 people.
Assessment

Assignment Sheets:

- An assignment sheet is usually made available on Monday on the module webpage.
- Solutions have to be handed in in the following week before the lecture on Monday.
- You can hand in your solutions by putting them in the mailbox "Efficient Algorithms" on the basement floor in the MI-building.
- Solutions have to be given in English.
- Solutions will be discussed in the tutorial of the week when the sheet has been handed in, i.e., sheet may not be corrected by this time.
- You can submit solutions in groups of up to 2 people.
Assessment

Assignment Sheets:

- Submissions must be handwritten by a member of the group. Please indicate who wrote the submission.

- Don’t forget name and student id number for each group member.
Assignment Sheets:

- Submissions must be handwritten by a member of the group. Please indicate who wrote the submission.
- Don’t forget name and student id number for each group member.
Assessment

Assignment can be used to improve your grade

If you obtain a bonus, your grade will improve according to the following function:

\[ f(x) = \begin{cases} 1 & \text{if } 1 \leq x \leq 4 \\ 0.3 & \text{or } 0.4, \text{ respectively.} \end{cases} \]

Examples:

- 3.3 \rightarrow 3.0
- 2.0 \rightarrow 1.7
- 3.7 \rightarrow 3.3
- 1.0 \rightarrow 1.0
- > 4.0 \text{ no improvement}
Assessment

Assignment can be used to improve your grade

- If you obtain a bonus your grade will improve according to the following function

\[
 f(x) = \begin{cases} 
 \frac{1}{10} \text{round} \left( 10 \left( \frac{\text{round}(3x) - 1}{3} \right) \right) & 1 < x \leq 4 \\
 x & \text{otw.}
\end{cases}
\]

- It will improve by 0.3 or 0.4, respectively.

Examples:

- 3.3 → 3.0
- 2.0 → 1.7
- 3.7 → 3.3
- 1.0 → 1.0
- > 4.0 no improvement
Assessment

Assignment can be used to improve your grade

- If you obtain a bonus your grade will improve according to the following function

\[ f(x) = \begin{cases} \frac{1}{10} \text{round} \left( 10 \left( \frac{\text{round}(3x) - 1}{3} \right) \right) & 1 < x \leq 4 \\ x & \text{otw.} \end{cases} \]

- It will improve by \(0.3\) or \(0.4\), respectively.

Examples:

- \(3.3 \rightarrow 3.0\)
- \(2.0 \rightarrow 1.7\)
- \(3.7 \rightarrow 3.3\)
- \(1.0 \rightarrow 1.0\)
- \(> 4.0\) no improvement
Assessment

Assignment can be used to improve your grade

- If you obtain a bonus your grade will improve according to the following function

\[
f(x) = \begin{cases} 
\frac{1}{10} \text{round} \left( 10 \left( \frac{\text{round}(3x) - 1}{3} \right) \right) & 1 < x \leq 4 \\
x & \text{otw.}
\end{cases}
\]

- It will improve by 0.3 or 0.4, respectively.
Examples:
- 3.3 → 3.0
- 2.0 → 1.7
- 3.7 → 3.3
- 1.0 → 1.0
- > 4.0 no improvement
Assessment

Assignment can be used to improve your grade

▶ If you obtain a bonus your grade will improve according to the following function

\[
f(x) = \begin{cases} 
\frac{1}{10} \text{round} \left( 10 \left( \frac{\text{round}(3x) - 1}{3} \right) \right) & 1 < x \leq 4 \\
x & \text{otw.}
\end{cases}
\]

▶ It will improve by 0.3 or 0.4, respectively.

Examples:

▶ 3.3 \rightarrow 3.0
▶ 2.0 \rightarrow 1.7
▶ 3.7 \rightarrow 3.3
▶ 1.0 \rightarrow 1.0
▶ > 4.0 no improvement
Assessment

Assignment can be used to improve your grade

- If you obtain a bonus your grade will improve according to the following function

\[
 f(x) = \begin{cases} 
 \frac{1}{10} \text{round} \left( 10 \left( \frac{\text{round}(3x) - 1}{3} \right) \right) & 1 < x \leq 4 \\
 x & \text{otw.} 
\end{cases}
\]

- It will improve by 0.3 or 0.4, respectively.

Examples:

- 3.3 → 3.0
- 2.0 → 1.7
- 3.7 → 3.3
- 1.0 → 1.0
- > 4.0 no improvement
Assessment

Assignment can be used to improve your grade

- If you obtain a bonus your grade will improve according to the following function

\[ f(x) = \begin{cases} 
\frac{1}{10} \text{round} \left( 10 \left( \frac{\text{round}(3x) - 1}{3} \right) \right) & 1 < x \leq 4 \\
x & \text{otw.}
\end{cases} \]

- It will improve by 0.3 or 0.4, respectively.

Examples:

- 3.3 \rightarrow 3.0
- 2.0 \rightarrow 1.7
- 3.7 \rightarrow 3.3
- 1.0 \rightarrow 1.0
- > 4.0 no improvement
Assessment

Assignment can be used to improve your grade

▶ If you obtain a bonus your grade will improve according to the following function

\[
f(x) = \begin{cases} 
\frac{1}{10} \text{round} \left( 10 \left( \frac{\text{round}(3x) - 1}{3} \right) \right) & 1 < x \leq 4 \\
x & \text{otw.}
\end{cases}
\]

▶ It will improve by 0.3 or 0.4, respectively.

Examples:

▶ 3.3 → 3.0
▶ 2.0 → 1.7
▶ 3.7 → 3.3
▶ 1.0 → 1.0
▶ > 4.0 no improvement
Assessment

Requirements for Bonus

- 50% of the points are achieved on submissions 2–8,
- 50% of the points are achieved on submissions 9–14,
- each group member has written at least 4 solutions.
1 Contents

- Foundations
  - Machine models
  - Efficiency measures
  - Asymptotic notation
  - Recursion

- Higher Data Structures
  - Search trees
  - Hashing
  - Priority queues
  - Union/Find data structures

- Cuts/Flows

- Matchings
1 Contents

- Foundations
  - Machine models
  - Efficiency measures
  - Asymptotic notation
  - Recursion

- Higher Data Structures
  - Search trees
  - Hashing
  - Priority queues
  - Union/Find data structures

- Cuts/Flows
- Matchings
1 Contents

- Foundations
  - Machine models
  - Efficiency measures
  - Asymptotic notation
  - Recursion
- Higher Data Structures
  - Search trees
  - Hashing
  - Priority queues
  - Union/Find data structures
- Cuts/Flows
1 Contents

- Foundations
  - Machine models
  - Efficiency measures
  - Asymptotic notation
  - Recursion

- Higher Data Structures
  - Search trees
  - Hashing
  - Priority queues
  - Union/Find data structures

- Cuts/Flows

- Matchings
2 Literatur

Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman: The design and analysis of computer algorithms, Addison-Wesley Publishing Company: Reading (MA), 1974


2 Literatur

- Ronald L. Graham, Donald E. Knuth, Oren Patashnik: *Concrete Mathematics*, 2. Auflage, Addison-Wesley, 1994
- Jon Kleinberg, Eva Tardos: *Algorithm Design*, Addison-Wesley, 2005
2 Literatur


Uwe Schöning: *Algorithmik*, Spektrum Akademischer Verlag, 2001