Marburg University

Department of Computer Science
Prof. Dr. Sebastian Wild

Date: 2025-10-16
Version: 2025-10-16 20:38

Übungsblatt 1 zur Vorlesung Effiziente Algorithmen (Winter 2025/26)

Abgabe: *Bis 2025-10-24 18:00, on ILIAS.*

Regeln und Leitlinien

- Die Übungsaufgaben sind integraler Bestandteil des Moduls. Sie greifen den Stoff aus der Vorlesung auf und vertiefen ihn.
- Für die Zulassung zur Prüfung benötigen Sie mindestens 50% der insgesamt erreichbaren Punkte, und nicht mehr als 2 Übungsblätter mit 0 Punkten.
- Sie dürfen in Gruppen von 2–4 Studierenden zusammenarbeiten und gemeinsame Lösungen abgeben.
- Wenn Sie externe Quellen verwenden, kennzeichnen Sie die übernommenen Teile eindeutig als solche, und geben Sie Referenzen zu den Originalquellen an. Bei Webseiten fügen Sie mindestens das Zugriffsdatum hinzu und verwenden Sie nach Möglichkeit dauerhafte Links (z. B. bei Wikipedia). Sie dürfen alle öffentlich zugänglichen Quellen zitieren, sollten jedoch deren Glaubwürdigkeit berücksichtigen: Sie sind dafür verantwortlich, die Richtigkeit der zitierten Inhalte zu überprüfen.

  Dies gilt umso mehr für Antworten aus generativer KI.

  Plagiate werden in der Wissenschaft nicht toleriert – weder aus externen Quellen, noch von anderen Gruppen oder KI-generierten Antworten; in schweren Fällen kann dies zum Ausschluss von der Universität führen.
- Die Aufgaben sind so gestaltet, dass sie ohne KI-Tools lösbar sind – berauben Sie sich nicht des Lerneffekts.
- Gemeinsames Arbeiten an *konzeptionellen* Lösungsansätzen zwischen Gruppen ist erlaubt, jede Gruppe muss jedoch eine eigenständige, detaillierte Ausarbeitung einreichen. Fügen Sie in solchen Fällen eine kurze Notiz hinzu, mit welchen Gruppen Sie bei einer bestimmten Aufgabe zusammengearbeitet haben.

1. Aufgabe

10 + 20 Punkte

Beweisen Sie folgende Aussagen durch Induktion.

a) Für alle $n \in \mathbb{N}_0$ gilt $\sum_{k=0}^{n} q^k = \frac{q^{n+1}-1}{q-1}$.

b) Für alle $n \in \mathbb{N}_0$ können die *Türme von Hanoi* mit n Scheiben in $2^n - 1$ Schritten gelöst werden.

   Eine anschauliche Beschreibung der Türme von Hanoi ist auf Wikipedia zu finden: `https://de.wikipedia.org/wiki/T%C3%BCrme_von_Hanoi`

   Tipp: Analysieren Sie einen rekursiven Algorithmus zur Lösung der Türme von Hanoi, der eine Turm der Höhe h von einer Start- zu einer Zielstange bewegt.

2. Aufgabe

20 Punkte

Das Ergebnis einer ganzzahligen Division hat zwei integrale Bestandteile: der *Quotient* und der *(ganzzahlige) Rest*. Für zwei ganze Zahlen n und $k > 0$ ist der Quotient das Ergebnis der ganzzahligen Divison „$n \operatorname{div} k$“ als die größte ganze Zahl m definiert, für die $m \cdot k \leq n$ gilt. Der Rest der Division ist dann $r = n - m \cdot k$. Dabei gilt stets $0 \leq r < k$. Wir bezeichnen r auch als das Ergebnis der Modulo-Operation, geschrieben „$r = n \bmod k$“.

Beispiel: $10 \operatorname{div} 3 = 3$ und $10 \bmod 3 = 1$,
$13 \operatorname{div} 5 = 2$ und $13 \bmod 5 = 3$.

Zeigen Sie mithilfe der *Potentialmethode*, dass die folgende Prozedur $Mod(n, k)$ stets terminiert, wenn sie mit Parametern $n \in \mathbb{N}_{\geq 1}$ und $k \in \mathbb{N}_{\geq 1}$ aufgerufen wird.

```
procedure Mod(n, k)
    // Input: positive integers n, k.
    // Output: value of n mod k.
    t := n
    while t ≥ k
        t := (t − k)
    end while
return t
```

3. Aufgabe

10 + 40 Punkte

Gegeben ist folgender Algorithmus, der die Potenz x^n für $n \in \mathbb{N}_0$ berechnet. Dabei ist “div” die ganzzahlige Division und “mod” der Rest der Division.

```
procedure power(x, n):
    // Input: x, n ∈ ℕ_0
    // Output: x^n
    p := 1
    if x ≠ 0
        while n ≠ 0
            if n mod 2 == 1
                p := p · x
            end if
            n := n div 2
            x := x · x
        end while
    else
        p := 0
    end if
    return p
```

a) Durchlaufen Sie den Algorithmus für folgende Werte: $x = 2$, $n = 7$. Geben Sie die Werte von x, n, p nach jedem Schleifendurchlauf an.

b) Verifizieren Sie die Korrektheit des Algorithmus mit Hilfe einer Schleifeninvariante. Geben Sie hierbei für jede Codestelle geeignete Vor- und Nachbedingungen an, und nennen Sie die Schleifeninvariante.