Übungen zur Vorlesung

Algorithmen und Datenstrukturen

Übungsblatt 1

Arbeitsgruppe Kryptographie und Komplexitätstheorie
Prof. Dr. Marc Fischlin
Dr. Christian Janson
Patrick Harasser
Felix Rohrbach

Sommersemester 2019
Veröffentlicht am: 19.04.2019

P1 (Gruppendiskussion)

Nehmen Sie sich etwas Zeit, um die folgenden Fachbegriffe in einer Kleingruppe zu besprechen, sodass sie anschließend in der Lage sind, die Begriffe dem Rest der Übungsgruppe zu erklären:

(a) Algorithmus
(b) Schleifeninvariante
(c) Totale Ordnung

P2 (Insertion Sort)

(a) In der Vorlesung haben Sie den Algorithmus Insertion Sort kennengelernt, der die Elemente einer Liste in aufsteigender Reihenfolge sortiert. Schreiben Sie diesen so um, dass er die Elemente in absteigender Reihenfolge sortiert. Beachten Sie, dass Ihr neuer Algorithmus weiterhin stabil bleiben soll.

(b) Sortieren Sie mithilfe dieses Algorithmus das folgende Array von Strings nach ihrer Länge: [“auf”,“Baum”,“Daten”, “Haus”,“sortieren”]. Geben Sie dabei die Zwischenschritte mit an.

P3 (Eigenschaften von Algorithmen)

Beschreiben Sie jeweils für die folgenden Algorithmen kurz, was diese berechnen, und prüfen Sie, welche Eigenschaften (Determiniertheit, Determinismus, Terminierung, Korrektheit, Effizienz) der Algorithmus erfüllt.

Algorithmus1(int[] array)

```
int n = array.length;
while true {
  bool sorted = true ;
  for (int i = 0; i < n − 1; i + +){
    if array[i] < array[i + 1]{
      sorted = false ;
    }
  }
  if sorted{
    return array;
  }
  int n1 = random.nextInt(n); // 0 ≤ n1 < n
  int n2 = random.nextInt(n);
  int tmp = array[n1];
  array[n1] = array[n2];
  array[n2] = tmp;
}
```

Algorithmus2(int n)

```
if n%2 == 0{
  return false
}
for (int i = 3; i < n; i = i + 2){
  if (n%i == 0){
    return false ;
  }
}
return true ;
```

P4 (Laufzeiten)

In der Tabelle sind verschiede Laufzeitfunktionen $f(n)$ abhängig von der Größe der Eingabe n in Millisekunden angegeben, sowie verschiedene Zeitabschnitte. Tragen Sie ein, wie groß n jeweils maximal sein darf, damit die Berechnung innerhalb des Zeitabschnitts durchläuft. Nehmen Sie einfachheitshalber an, dass ein Tag aus 24 Stunden und ein Jahr aus 365 Tagen besteht.

	1 Sekunde	1 Minute	1 Stunde	1 Tag	1 Monat	1 Jahr	1 Jahrhundert
$\mathrm{sqrt}(n)$							
n							
$n \log_2(n)$							
n^2							
n^3							
2^n							
$n!$							

P5 (Türme von Hanoi)

Die *Türme von Hanoi* sind ein bekanntes Geduldsspiel, dessen Regeln hier nochmal kurz erklärt werden.

Das Spiel besteht aus drei Stäben. Auf die Stäbe wird eine feste Anzahl gelochter Scheiben gelegt, alle unterschiedlicher Größe. Zu Beginn liegen alle Scheiben auf einem Stab, der Größe nach geordnet, mit der größten Scheibe ganz unten. Ziel des Spiels ist es, alle Scheiben vom Ausgangsstab auf einen anderen Stab zu versetzen. Dabei muss folgende Regel beachtet werden: Bei jedem Zug darf immer nur die oberste Scheibe eines beliebigen Stapels bewegt, und auf einen anderen Stab gelegt werden, unter der Voraussetzung, dass sich dort nicht schon eine kleinere Scheibe befindet.

(a) Geben Sie einen rekursiven Algorithmus an, der die Lösung des Spiels beschreibt, und berechnen Sie dessen Laufzeit.
(b) Einer Legende zufolge gibt es irgendwo in der Stadt Hanoi einen Brahma-Tempel, in dem Mönche dieses Spiel mit 64 Scheiben aus purem Gold spielen. Die Legende besagt, dass die Welt in Schutt und Asche fallen wird, sobald sie mit dem Spiel fertig sind. Angenommen, die Mönche verwenden Ihren Algorithmus aus (a) und verschieben eine Scheibe pro Sekunde, wie lange wird es dauern, bis sie das Spiel zu Ende gespielt haben? Geben Sie eine grobe Abschätzung.

(c*) *Zusatzaufgabe:* Zeigen Sie, dass Ihr Algorithmus aus (a) "optimal" ist: Es gibt keinen korrekten Algorithmus der es ermöglicht, das Spiel in weniger Schritten zu lösen. (Falls dies nicht der Fall sein sollte, revidieren Sie Ihre Angabe.) Was bedeutet das für die Mönche aus (b)? Müssen wir uns über den "Weltuntergang durch die Türme von Hanoi" Gedanken machen?[1]

[1] Unser Universum ist ca. 13,8 Milliarden Jahre alt.