

Lösungsvorschläge zur Probeklausur zu Algorithmen und Datenstrukturen

Aufgabe 1 Rechnen mit Landau-Symbolen

(5 Punkte)

Untersuchen Sie, ob folgende Behauptung wahr ist, und begründen Sie Ihre Aussage unter Angabe passender Konstanten.

$$-\frac{4}{n^2} + \frac{5}{n} + 5 = O(1)$$

Zu zeigen ist, dass ein $n_0 \in \mathbb{N}$ und ein $c \in \mathbb{R}$ existieren so dass für alle $n \geq n_0$ gilt:

$$-\frac{4}{n^2} + \frac{5}{n} + 5 \leq c$$

Wahl der Konstanten: wähle z.B. $n_0 = 1$ und ein beliebiges $c \geq 6$ (also z.B. $c = 7$). Alternativ wähle ein $n_0 > 1$ und wähle entsprechend ein passendes $c \geq -\frac{4}{n_0^2} + \frac{5}{n_0} + 5$ (z.B. $n_0 = 5$ und $c = 6$).

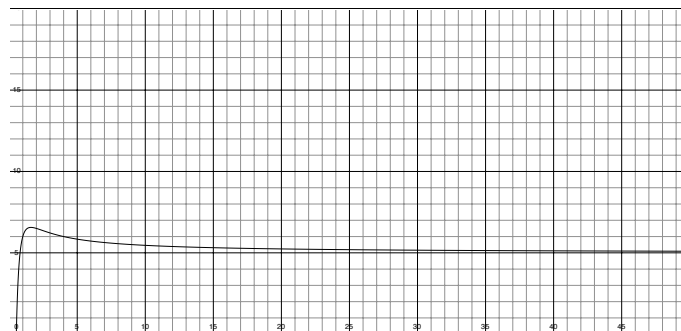
Mit diesen Konstanten gilt die Ungleichung, und die Behauptung ist gezeigt. \square

Alternative Schlussfolgerung, dass Behauptung gilt: Die Folgen $1/n^i$ konvergieren gegen 0, daher gilt die obige Ungleichung. (korrekte Konstanten sind trotzdem anzugeben!)

[[Optional: Bestimmung des Maximums der Funktion links durch Ableitung der linken Funktion auf Null setzen

$$\frac{8}{n} - 5 = 0 \Rightarrow n = 1.6$$

ergibt ein Maximum der Funktion für $n = 1.6$ mit einem Funktionswert von $6\frac{9}{16} = 6.5625$. Siehe auch Plot unten.]]



Bewertung:

- 1 Punkt für Ungleichung $O(1)$.
- 2 Punkte für korrekte Wahl n_0, c .
- 1 Punkt für korrekte Schlussfolgerung.

Aufgabe 2 Boolesche Logik

(7 Punkte)

Gegeben sei die Funktion:

$$f : \begin{cases} \mathbb{B} \times \mathbb{B} \times \mathbb{B} & \rightarrow & \mathbb{B} \\ (x_0, x_1, x_2) & \mapsto & \begin{cases} 1, & \text{falls Binärzahl } [x_2x_1x_0]_2 \text{ keine Primzahl ist} \\ 0, & \text{sonst} \end{cases} \end{cases}$$

Zeigen Sie mittels einer Wahrheitstabelle, wie in Vorlesung und Übung behandelt, dass der folgende logische Ausdruck gilt:

$$f(x_0, x_1, x_2) = (\neg x_0 \wedge \neg x_1 \wedge \neg x_2) \vee (x_0 \wedge \neg x_1 \wedge \neg x_2) \vee (\neg x_0 \wedge \neg x_1 \wedge x_2) \vee (\neg x_0 \wedge x_1 \wedge x_2)$$

Geben Sie dabei für jeden durch Klammern umschlossenen Term eine eigene Spalte an!

Hinweis: Eine Primzahl ist eine natürliche Zahl $n \in \mathbb{N}, n > 1$, die nur durch sich selbst und durch 1 ganzzahlig teilbar ist.

x_0	x_1	x_2	T_1 $\neg x_0 \wedge \neg x_1 \wedge \neg x_2$	T_2 $x_0 \wedge \neg x_1 \wedge \neg x_2$	T_3 $\neg x_0 \wedge \neg x_1 \wedge x_2$	T_4 $\neg x_0 \wedge x_1 \wedge x_2$	$T_1 \vee T_2 \vee T_3 \vee T_4$	$[x_2x_1x_0]_2$	$f(x_0, x_1, x_2)$
0	0	0	1	0	0	0	1	0 ₁₀	1
1	0	0	0	1	0	0	1	1 ₁₀	1
0	1	0	0	0	0	0	0	2 ₁₀	0
1	1	0	0	0	0	0	0	3 ₁₀	0
0	0	1	0	0	1	0	1	4 ₁₀	1
1	0	1	0	0	0	0	0	5 ₁₀	0
0	1	1	0	0	0	1	1	6 ₁₀	1
1	1	1	0	0	0	0	0	7 ₁₀	0

Die den beiden Seiten der Gleichung entsprechenden Spalten sind gleich, damit ist die Behauptung gezeigt. □

Alternativtabelle:

x_0	x_1	x_2	T_1 $\neg x_0 \wedge \neg x_1 \wedge \neg x_2$	T_2 $x_0 \wedge \neg x_1 \wedge \neg x_2$	T_3 $\neg x_0 \wedge \neg x_1 \wedge x_2$	T_4 $\neg x_0 \wedge x_1 \wedge x_2$	$T_1 \vee T_2 \vee T_3 \vee T_4$	$[x_2x_1x_0]_2$	$f(x_0, x_1, x_2)$
0	0	0	1	0	0	0	1	0 ₁₀	1
0	0	1	0	0	1	0	1	4 ₁₀	1
0	1	0	0	0	0	0	0	2 ₁₀	0
0	1	1	0	0	0	1	1	6 ₁₀	1
1	0	0	0	1	0	0	1	1 ₁₀	1
1	0	1	0	0	0	0	0	5 ₁₀	0
1	1	0	0	0	0	0	0	3 ₁₀	0
1	1	1	0	0	0	0	0	7 ₁₀	0

Bewertung:

- 1 Punkt für Aufstellung der Wahrheitstabelle mit x_0, x_1, x_2 .
- 4 Punkte für Min-Term-Spalten T_i (jeweils 1 Punkt pro Spalte).
- 1 Punkt für Disjunktions-Spalte $T_1 \vee T_2 \vee T_3 \vee T_4$.
- 1 Punkt für Spalte $f(x_0, x_1, x_2)$.
- Alle anderen Spalten dienen der Illustration und ergeben keine Punkte!

Aufgabe 3 Zahldarstellung

(4 Punkte)

Konvertieren Sie die Zahl 7_{10} in eine 4-bit Binärzahl. Berechnen Sie dann das 2-Komplement dieser 4-bit Binärzahl. Warum lässt sich das entstandene Bitmuster in der 2-Komplement-Darstellung als negative Zahl interpretieren?

Konversion in Binärzahl:

$$7_{10} = 4_{10} + 2_{10} + 1_{10} = 0111_2$$

Zweierkomplement ist bitweise Inversion und Addition von 1:

$$\overline{0111}_2 + 1_2 = 1000_2 + 1_2 = 1001_2$$

Das erste Bit stellt das Vorzeichen dar: 1001_2 kann interpretiert werden als

- 0111_2 ist positiv (wegen 0): also 7_{10}
- 1001_2 ist negativ (wegen 1), also -7_{10}

Bewertung: 1 Punkt für korrekte 4-bit Binärzahl, 1 Punkt für korrektes 2-Komplement, jeweils 1 Punkt für korrekte Interpretation des Bitmusters.

Aufgabe 4 Sequentielle Liste

(2 Punkte)

Gegeben sei ein Feld A implementiert als sequentielle Liste (Array). Was ist die Komplexität der Einfüge-Operation `insert` und der Zugriffs-Operation `elementAt`?

insert : Man muss Elemente hinter der Einfügeposition nach hinten verschieben, diese Operation hat die Komplexität $O(n)$.

elementAt : In einer sequentiellen Liste hat man direkten Zugriff auf die einzelnen Elemente, daher ist die Komplexität $O(1)$.

Bewertung: Jeweils 1 Punkt für korrekte Komplexität.

Aufgabe 5 Stack

(3 Punkte)

Nennen Sie kurz *eine* Methode, um den abstrakten Datentyp Stack mittels elementarer Datenstrukturen zu implementieren. Was sind dann jeweils die Komplexitäten der Operation `push` und `pop`?

Möglichkeit 1: Einfach verkettete Liste. Der Zeiger `start` zeigt hierbei auf das oberste Element des Stacks.

Möglichkeit 2: Sequentielle Liste mit Index-Variable.

Komplexität von `push` und `pop` bei beiden Möglichkeiten jeweils $O(1)$, da die Liste nie durchlaufen werden muss.

Bewertung: 1 Punkt für korrekte Implementierung von Stack mittels elementarer Datenstruktur, jeweils 1 Punkt für korrekte Komplexitäten.

Aufgabe 6 Verifikation

(6 Punkte)

Gegeben sei ein Feld von reellen Zahlen $X[0..n-1]$ der Länge n . Die Funktion **Mean**(X) soll den Mittelwert \bar{X} von X berechnen,

$$\bar{X} = \frac{1}{n} \sum_{i=0}^{n-1} X[i]$$

Nachfolgend der Pseudocode der Funktion **Mean**(X). Hierbei sind die in geschweiften Klammern eingeschlossenen Bezeichner $\{C_k\}$, wie aus Vorlesung und Übung gewohnt, Platzhalter für logische Bedingungen, die zur Verifikation benötigt werden.

Mean(X):

$\{C_0\}$

`meanval` = 0;

`i` = 0;

$\{C_1\}$

while (`i` < `n`) {

$\{C_2\}$

`meanval` = `meanval` + $\frac{X[i]}{n}$;

`i` = `i` + 1;

$\{C_3\}$

 }

return `meanval`;

$\{C_4\}$

- a) Geben Sie jeweils eine geeignete Vor- und Nachbedingung C_0 bzw. C_4 der Funktion **Mean**(X) an!

$$\begin{aligned}C_0 &= (n > 0 \wedge X[k] \in \mathbb{R} \forall k \in \{0 \dots n-1\}) \\C_4 &= (\bar{X} = \text{meanval})\end{aligned}$$

Bei C_0 kommen meanval , i nicht vor.

Bei C_4 ist $\text{mean}(X) = \text{meanval}$ sinnlos!

Bewertung: jeweils 1 Punkt für korrekte Vor- und Nachbedingung.

- b) Geben Sie eine geeignete Schleifeninvariante P für die Schleife **while** ($i < n$) { ... } an!
Was ist die Vorbedingung C_1 dieser Schleife?

$$\begin{aligned}P &= (\text{meanval} + \frac{1}{n} \sum_{j=i}^{n-1} X[j] = \bar{X}) \\C_1 &= C_0 \wedge (i = 0) \wedge (\text{meanval} = 0)\end{aligned}$$

Bewertung: 1 Punkt für korrekte Schleifeninvariante, 1 Punkt für korrekte Vorbedingung.

- c) Welche drei Bedingungen müssen überprüft werden, damit die Korrektheit der **while** Schleife mittels der Invariante P gezeigt werden kann? Zeigen Sie, dass diese drei Bedingungen in diesem Fall erfüllt sind!

Hinweis: Sie können hierzu die Vorbedingungs-Platzhalter C_1, C_2, C_3, C_4 verwenden.

Zu zeigen sind:

- **Bedingung 1: Schleifeneintritt:** $C_1 \Rightarrow P$
Trivial, es gilt $i = 0$ und $\text{meanval} = 0$, damit entspricht P der Definition des Mittelwerts.
- **Bedingung 2: Schleifenkörper:** $\{C_2 = P \wedge (i < n)\} \dots \{C_3 = P\}$
Wir schreiben die neuen Werte mit Hilfsvariablen:

$$\begin{aligned}\text{meanval}' &= \text{meanval} + \frac{X[i]}{n} \\i' &= i + 1\end{aligned}$$

Soll bei C_3 die Invariante P gelten, so betrachten wir

$$\begin{aligned}\text{meanval}' + \frac{1}{n} \sum_{j=i'}^{n-1} X[j] &= \text{meanval} + \frac{X[i]}{n} + \frac{1}{n} \sum_{j=i+1}^{n-1} X[j] \\&= \text{meanval} + \frac{1}{n} \sum_{j=i}^{n-1} X[j] \\&= \bar{X}\end{aligned}$$

Dabei gilt die letzte Zeile, da zuvor die Invariante P ja erfüllt war, und insgesamt gilt P auch am Ende des Schleifenkörpers.

- **Bedingung 3: Schleifenende:** $(P \wedge \neg(i < n)) \Rightarrow C_4$
Wir betrachten den ersten Fall mit $\neg(i < n)$, also $i = n$. Dann gilt:

$$P \wedge (i = n) \Rightarrow \text{meanval} = \bar{X}$$

Dies ist genau die Nachbedingung C_4 !

Bewertung: jeweils 1 Punkt für Bedingungen, jeweils 1 Punkt für korrekte Anwendung, insgesamt 6 Punkte.

Aufgabe 7 Sortieren

(7 Punkte)

- a) Geben Sie einen Sortier-Algorithmus an, der im Mittel mit Komplexität $O(n \log n)$ sortiert! Nach welchem Algorithmen-Muster wurde dieser Algorithmus entworfen?

Divide and Conquer: Merge oder Quick Sort

Bewertung: 1 Punkt für Merge oder Quick Sort, 1 Punkt für Divide and Conquer.

- b) Gegeben sei eine bereits sortierte Liste von natürlichen Zahlen der Länge n . In diese Liste soll nun ein weiteres Element einsortiert werden. Schlagen Sie einen möglichst effizienten Algorithmus für diese Aufgabe vor und geben Sie dessen Komplexität an.

Insertion Sort ohne die äußere Schleife, damit dann $O(n)$.

Bewertung: 1 Punkt für Insertion Sort, 1 Punkt für Komplexität.

- c) Skizzieren Sie (durch Angabe des Zustandes nach jedem Durchlauf der Hauptschleife), wie der *Quick Sort* die folgende Zahlenreihe sortieren würde! Wählen Sie als Pivot-Element stets das erste Element.

5 , 7 , 1 , 9 , 3

(Aktive Pivot-Elemente sind grün eingefärbt, während bearbeitete gelb eingefärbt wurden)

5 , 7 , 1 , 9 , 3

3 , 7 , 1 , 9 , 5

3 , 1 , 5 , 9 , 7

3 , 1 , 5 , 9 , 7

1 , 3 , 5 , 7 , 9

1 , 3 , 5 , 7 , 9

1 , 3 , 5 , 7 , 9

Bewertung: 0,5 Punkte pro Schritt, letzter Schritt kann entfallen (da trivial).

In einer echten Klausur sollten Sie von den erreichbaren Punkten mindestens die Hälfte erzielen, in diesem Beispiel also 17 Punkte.