

1. Übungsblatt für Track ε zur Vorlesung Entwurf und Analyse von Algorithmen, WS 13/14

Abgabe: Bis Freitag, 25.10.2013, 12:00 Uhr, Abgabekasten im Treppenhaus 48-6.

Übungskonzept

Eine erfolgreiche Bearbeitung der Übungsaufgaben ist Zulassungsvoraussetzung für die Abschlussklausur. Es gelten folgende Regeln:

- Die Übungen finden nach Tracks getrennt statt. Dieses und folgende Blätter im Track ε sind für Studenten aus den Studiengängen „Informatik“ und „Mathematik“ vorgesehen.
- Es gibt *Basis-* und *Aufbauaufgaben*. Von ersteren muss jeder Student *einzel*n je eine pro Woche abgeben, letztere werden in *Vierergruppen* abgegeben. Pro Vierergruppe darf jede Basisaufgabe nur einmal abgegeben werden.
- Für die Zulassung sind 50 % der mittleren erreichbaren Basis- und 20 % aller Aufbaupunkte nötig.

Beachten Sie, dass sich Ihre relative Basispunktzahl $B = B_s/B_{\max}$ aus den von Ihnen erreichten Basispunkten B_s und aus den maximalen Punktzahlen $B_{i,j}$ der k_j Basisaufgaben auf Blatt j so berechnet:

$$B_{\max} = \sum_{j=1}^{13} \frac{1}{k_j} \cdot \sum_{i=1}^{k_j} B_{i,j} .$$

Sie benötigen also $B \geq 1/2$.

- Weiterhin ist für Informatiker eine Zwischenklausur zu bestehen, für Mathematiker ist die Teilnahme freiwillig. In der Zwischenklausur werden außerdem Bonusbasispunkte zu erwerben sein.

Details finden Sie auf der Website zur Vorlesung:

<http://wwwagak.cs.uni-kl.de/Vorlesung/ea1314.html>

Basisaufgaben

B1: WHILE-Programme

3 Punkte

Für alle Programmieraufgaben gilt, dass (Teil-)Programme, die bereits in der Vorlesung, in *früheren* Übungen oder im Buch¹ (Abschnitt 1.2) erarbeitet wurden, als Makros verwendet werden können. Bitte versehen Sie Ihre Programme mit *für das Verständnis hilfreichen* Kommentaren!

Beachten Sie außerdem, dass Teilprogramme, die als Makros verwendet werden sollen – insbesondere die in dieser Aufgabe gefragten – natürlich die Eingabevariablen *nicht* überschreiben sollten, außer der Seiteneffekt ist gewünscht. Zum Beispiel sollte bei einem Makro für `x := x div z` die Variable `z` unverändert bleiben, während `x` sich natürlich ändert².

Benötigen Sie in einem (Teil-)Programm eine neue Variable, können Sie stets annehmen, dass eine solche existiert (und mit 0 initialisiert ist). Machen Sie bitte kenntlich, wenn Sie dies benutzen.

Im Folgenden seien x_i und x_j Variablen und c eine Konstante.

- Geben Sie ein WHILE-(Teil-)Programm an, das die Zuweisung $x_j := c - x_i$ ausführt.
- Geben Sie ein WHILE-(Teil-)Programm an, das `If $x_i = c$ THEN $x_j := 0$ ELSE $x_j := x_j + 1$ END` berechnet.
- Geben Sie ein WHILE-(Teil-)Programm an, das die Potenz $x_i^{x_j}$ berechnet.

B2: Wahrscheinlichkeitsrechnung

3 Punkte

Wir betrachten die Menge $\Omega = \{1, 2, \dots, n\}$. In einem ersten Zufallsexperiment bestimmen wir eine Menge $A \subseteq \Omega$, indem wir jedes Element aus Ω unabhängig von den anderen mit Wahrscheinlichkeit p_A in A aufnehmen. Wir wiederholen das Experiment und bilden eine Menge B , wobei wir jedes Element aus Ω (wieder unabhängig von den anderen) mit Wahrscheinlichkeit p_B in B aufnehmen.

- Bestimmen Sie $\mathbb{E}[|A \cap B|]$, den Erwartungswert der Mächtigkeit der Schnittmenge.
- Bestimmen Sie $\mathbb{E}[|A \cup B|]$, den Erwartungswert der Mächtigkeit der Vereinigungsmenge.

¹Unter “dem Buch” verstehen wir im Kontext dieser Vorlesung immer:

Markus Nebel, *Entwurf und Analyse von Algorithmen*, Springer Vieweg Wiesbaden, 2012, ISBN: 978-3-8348-1949-9

²Sie sehen schon, dass Sie im Sinne besserer Lesbarkeit natürlich Ihre Variablen beliebig benennen können – solange klar ist, welche Variable welche Rolle einnimmt (also etwa Eingabe und Ausgabe).

B3: WHILE-Programme

3 Punkte

Geben Sie ein WHILE-Programm an, das $\frac{x_i}{x_j}$ mit drei Nachkommastellen berechnet, wobei die Nachkommastellen für die Ausgabe an die Vorkommastelle konkateniert werden sollen. (Also würde das Ergebnis 3,725 als 3725 ausgegeben werden.)

B4: Wahrscheinlichkeitenerzeugendenfunktionen

3 Punkte

Eine Zufallsvariable X mit Werten in \mathbb{N} heißt *verschoben geometrisch* verteilt mit Parameter $p \in (0, 1)$, wenn für alle $k \in \mathbb{N}$

$$\Pr[X = k] = (1 - p)^{k-1} \cdot p$$

gilt. Anschaulich beschreibt X die Anzahl der Wiederholungen, bis bei der wiederholten Ausführung eines Bernoulli-Zufallsexperiments ein Ereignis mit Wahrscheinlichkeit p das erste Mal auftritt.

Bestimmen Sie die geschlossene Form der Wahrscheinlichkeiten-Erzeugendenfunktion von X und nutzen Sie diese, um den Erwartungswert $\mathbb{E}[X]$ in Abhängigkeit von p zu berechnen.

Aufbauaufgaben**1. Aufgabe**

5 + 4 Punkte

- a) Sei X eine Zufallsvariable mit $p_n := \Pr[X = n]$, wobei $\sum_{k \in \mathbb{N}_0} p_k = 1$ gelte. Zeigen Sie, dass für $P(z)$ die zugehörige Wahrscheinlichkeiten-Erzeugendenfunktion die Varianz von X über

$$\mathbb{V}[X] = P''(1) + P'(1) - (P'(1))^2$$

berechnet werden kann. Hierbei bezeichnen $P'(z)$ bzw. $P''(z)$ die erste bzw. zweite Ableitung von P nach z .

- b) Sei X eine Zufallsvariable mit Werten in \mathbb{N}_0 und Wahrscheinlichkeitsgewichten

$$\Pr[X = k] = (m + 1)^{-n} \binom{n}{k} m^{n-k}$$

für ganzzahlige Konstanten $n, m \geq 1$. Berechnen Sie – Sie ahnen es: mit Wahrscheinlichkeiten-Erzeugendenfunktionen – Erwartungswert $\mathbb{E}[X]$ und Varianz $\mathbb{V}[X]$ von X .

Hinweis: Für die Bearbeitung dieser Aufgaben mag das *Theoretical Computer Science Cheat Sheet*³ hilfreich sein.

2. Aufgabe

3 + 2 Punkte

Gegeben sei folgende Java-Methode:

```
1 int countDivisors(int n) {
2     if ( n <= 1 ) {
3         return 1;
4     }

6     int s = 1;
7     while ( s*s <= n ) {
8         ++s;
9     }
10    --s;

12    int nDivisors = 2;
13    int i;
14    for ( i = 2; i <= s; ++i ) {
15        nDivisors += n % i == 0 ? 2 : 0;
16    }
17    --i;

19    if ( i*i == n ) {
20        --nDivisors;
21    }

23    return nDivisors;
24 }
```

Hinweis: *condition ? value : value* ist der sogenannte conditional operator oder Elvis ?: operator, siehe z. B. http://en.wikipedia.org/wiki/Conditional_operator.⁴

- Übersetzen Sie die Java-Methode in ein äquivalentes Loop/While-Programm. Nur die im Buch (Abschnitt 1.2) erwähnten und in Aufgabe B1 implementierten Abkürzungen dürfen verwendet werden. Kommentieren Sie Ihr Programm sinnvoll!
- Kann die obige Java-Methode in ein reines Loop-Programm übersetzt werden? Begründen Sie (informell) warum bzw. warum nicht.

³<http://www.tug.org/texshowcase/cheat.pdf>

⁴Man beachte, dass Wikipedia hier nicht als wissenschaftliche Quelle verwendet ist, sondern nur eine allgemeine Einführung bietet. ;)