Mathematik für Informatiker
Kombinatorik, Stochastik und Statistik
Übungsblatt 7

Abgabe am Donnerstag, den 19.06.2025 bis 11:59 in OpenOlat.

1. In einem Glücksspiel würfeln wir m-mal mit einem Würfel mit n Seiten. Wir gewinnen, falls wenigstens eine 1 auftritt, ansonsten verlieren wir. Zeigen Sie:

   (a) Die Wahrscheinlichkeit in dem Spiel keine 1 zu würfeln ist

   $$\left(1-\frac{1}{n}\right)^m$$

   (b) Für $m = n \cdot \ln(2)$ Würfe ist das Spiel im Grenzwert $n \to \infty$ fair. Zeigen Sie dazu, dass

   $$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n\cdot\ln(2)} = \frac{1}{2}.$$

   Hinweise:

   - Für $x > 0$ und $a \in \mathbb{R}$ ist

     $$x^a := \exp(a \cdot \ln(x)).$$

   - Verwenden Sie die Regel von l'Hospital.

2. Wir wollen die Menge

   $$X = \{56, 64, 58, 61, 75, 86, 17, 62, 8, 50, 87, 99, 67, 10, 74\} \subset \mathbb{Z}$$

   mit $n = 15$ Elementen mit Hilfe des Quicksort-Algorithmus sortieren.

   (a) Finden Sie einen Baum von Pivotelementen, sodass der Algorithmus angewendet auf X mindestens $\binom{n}{2}$ Vergleiche benötigt.

   (b) Durch welchen Baum von Pivotelementen können Sie weniger als $n \cdot \ln(n)$ Vergleiche erreichen?

   Hinweis: Als Laufzeit bezeichnen wir hier die insgesamt notwendige Anzahl von Vergleichen.

3. Der Stupid-Sort-Algorithmus überprüft, ob eine Liste mit n paarweise verschiedenen Elementen sortiert ist. Falls nicht, permutiert er die Liste zufällig und prüft erneut. Berechne die Wahrscheinlichkeit, dass eine ursprünglich nicht sortierte Liste nach m Iterationen noch nicht sortiert ist.

4. Sei Ω eine endliche Menge, und $m : \Omega \to \mathbb{R}_{\geq 0}$ eine Wahrscheinlichkeitsfunktion. Die Wahrscheinlichkeit einer Teilmenge $M \subset \Omega$ definieren wir als

$$P(M) = \sum_{\omega \in M} m(\omega).$$

Zeigen Sie, dass für $M_1, \ldots, M_n \subset \Omega$ gilt

$$P(M_1 \cup \ldots \cup M_n) = \sum_{k=1}^{n} (-1)^{k-1} \sum_{|T|=k} P(M_T)$$

mit

$$M_T = \bigcap_{i \in T} M_i$$

für $T \subset \{1, ..., n\}$.

Hinweis: Folgen Sie dem Beweis der Siebformel.

5. (4 Zusatzpunkte)

   (a) Implementieren Sie den randomisierten Quicksort-Algorithmus.

   (b) Erproben Sie die asymptotische Laufzeit Ihrer Implementierung, indem Sie für verschiedene n eine Menge von zufällig erzeugten Zahlen $M = \{x_1, \ldots, x_n\} \subset \mathbb{Z}$ sortieren.

   Hinweise:

   - Die Laufzeit messen wir, indem wir in der Implementierung die Anzahl der durchgeführten Vergleiche zählen.
   - Die MAPLE-Funktion `rand(m)()` liefert eine Zufallszahl in $\{0, \ldots, m-1\}$.