Maximilian Katzmann
Hans-Peter Lehmann
Stefan Walzer

Übungsblatt 09

Randomisierte Algorithmik – Wintersemester 2023/2024

Aufgaben zur Vorlesung vom 21.12.2023
Abgabe im ILIAS bis 11.1.2024, 11:30 Uhr
Besprechung am 16.01.2024, 08:00 Uhr
Achte insbesondere bei handschriftlichen Abgaben auf Lesbarkeit. Die Abgabe erfolgt über das Übungsmodul im ILIAS. Gib Deine Ausarbeitungen in *einer* PDF-Datei ab.

Aufgabe 1 – Yaos Prinzip ohne Schnick-Schnack(-Schnuck)

Beweise Yaos Prinzip ohne auf spieltheoretische Sätze zurückzugreifen (kein Satz von Nash, Loomis, etc). Beweise also, dass im Setting der Vorlesung für eine beliebige Verteilung $\mathcal{A}_0$ auf **Algos** und eine beliebige Verteilung $\mathcal{I}_0$ auf **Inputs** gilt:

$$\max_{I \in \textbf{Inputs}} \mathbb{E}_{A \sim \mathcal{A}_0}[C(A, I)] \geq \min_{A \in \textbf{Algos}} \mathbb{E}_{I \sim \mathcal{I}_0}[C(A, I)].$$

Hinweis: Der spieltheoretische Vorbau der Vorlesung ist hier nicht erforderlich, weil wir nicht zeigen möchten, dass „=“ möglich ist. Es ist nützlich die erwarteten Kosten von $\mathcal{A}_0$ für die Eingabeverteilung $\mathcal{I}_0$ in den Blick zu nehmen.

Aufgabe 2 – Untere Schranken fürs Sortieren

Sei $n \in \mathbb{N}$ und **Inputs** die Menge aller Permutation von $\{1, \ldots, n\}$, das heißt die Menge aller Folgen von genau n Zahlen, die jede Zahl von $\{1, \ldots, n\}$ genau einmal enthalten[1]. Sei **Algos** die Menge aller vergleichsbasierten *deterministischen* Sortieralgorithmen.

Klarstellung des Modells. *Vergleichsbasiert* bedeutet, dass ein Algorithmus A bei Eingabe I wiederholt zwei Indizes i, j nennen kann und dann erfährt ob $I[i] < I[j]$ gilt oder nicht. Die Kosten $C(A, I)$ sind die Anzahl derartiger Vergleiche, die A auf I macht. *Sortieralgorithmus* bedeutet, dass der Algorithmus das Eingabearray durch „swaps“ modifiziert, um am Ende $[1, 2, \ldots, n]$ zu erhalten. Der Algorithmus darf nicht auf andere als die beschriebene Weise mit dem Array interagieren.[2]

[1] Zum Beispiel ist **Inputs** $= \{[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]\}$ für $n = 3$.

[2] Der Algorithmus darf sich zum Beispiel nicht die Bitdarstellung der Zahlen anschauen. Er darf auch nicht einfach das Eingabearray ignorieren und ein neues Array mit Inhalt $[1, 2, \ldots, n]$ zurückgeben.

(i) Nenne ein $A \in \textbf{Algos}$ (ohne Beweis) der erfüllt:

$$\max_{I \in \textbf{Inputs}} C(A, I) = O(n \log n).$$

Wie du sicher weist, gibt es keinen deterministischen Algorithmus, der Worst-Case Kosten $o(n \log n)$ erreicht. Wir erinnern uns an ein Resultat von Übungsblatt 1, Aufgabe 1, wo für $n = 3$ ein randomisierter Algorithmus $\mathcal{A}$ formuliert wurde, der den besten deterministischen Algorithmus schlägt:

$$\min_{A \in \textbf{Algos}} \max_{I \in \textbf{Inputs}} C(A, I) = 3 > 2 + \tfrac{2}{3} = \max_{I \in \textbf{Inputs}} \mathbb{E}_{A \sim \mathcal{A}}[C(A, I)].$$

Unser Plan im Folgenden ist zu zeigen, dass dieser Vorteil in O-Notation verschwindet, dass also für die randomisierte Komplexität des Sortierproblems gilt:

$$C := \min_{\mathcal{A} \text{ Vert. auf } \textbf{Algos}} \max_{I \in \textbf{Inputs}} \mathbb{E}_{A \sim \mathcal{A}}[C(A, I)] = \Omega(n \log n).$$

(ii) Zeige, dass es keine „schwierige Eingabe“ gibt, dass nämlich gilt:

$$\max_{I \in \textbf{Inputs}} \min_{A \in \textbf{Algos}} C(A, I) = n - 1.$$

Um gleich die bestmöglichen Kosten für eine Eingabe*verteilung* in den Blick nehmen, benötigen wir etwas Vorbeitung:

(iii) Zeige, dass in einem Binärbaum mit k Blättern die durchschnittliche Tiefe eines Blattes mindestens $\lfloor \log_2(k) \rfloor$ beträgt.
Hinweis: Zeige dafür, dass die durchschnittliche Blatttiefe bei balancierten Bäumen minimal ist, genauergesagt, dass sich jeder Baum, in dem sich die Blatttiefen um mindestens 2 unterscheiden, so umbauen lässt, dass die durchschnittliche Blatttiefe sinkt.

(iv) Folgere nun mithilfe von Yaos Prinzip, dass gilt:

$$C = \Omega(n \log n).$$

Hinweis: Als Verteilung auf den Eingaben bietet sich hier die uniforme Verteilung an.

Aufgabe 3 – Empfehlung: Simulating the Evolution of Teamwork

Zwei Tage nach der Vorlesung ist ein neues Video auf dem Youtube-Kanal *Primer* erschienen. Dort geht es unter anderem darum, alle möglichen 2-Spieler-Spiele mit zwei reinen Strategien danach zu klassifizieren, wieviele und welche Arten von Nash-Equilibria für diese existieren. Das Video ist unterhaltsam und lädt zum mitdenken ein, ist aber nur bedingt vorlesungsrelevant.

https://www.youtube.com/watch?v=TZfh8hpJIxo