Algorithmen und Datenstrukturen 2

ALGO2 · WiSe-2021/22 · tcs.uni-frankfurt.de/algo2 · 2021-11-19 · *4b83816*

Übungen zu Woche 6: Randomisierte Algorithmen II

Das Übungsblatt enthält alle empfohlenen Lernaktivitäten für die aktuelle Woche.

- **Heimarbeit bis Montag 17:00.**
  - Schau die Videos an und lies die Buchkapitel.
  - Bearbeite die 🌱-Aufgabe in Moodle. (Feste Abgabefrist!)
  - Lese den Aufgabentext aller Übungsaufgaben.
- **Heimarbeit.** Bearbeite die Übungsaufgaben soweit möglich. Probier zumindest alle mal!
- **Dienstag/Donnerstag.**
  - **8:00–8:15.** Besprechung im Hörsaal.
  - **8:15–9:15.** Bearbeite jetzt die Übungen, die du noch nicht lösen konntest. Sprich mit anderen Studis! Frag das Vorlesungsteam um Hilfe!
  - **9:15–9:45.** Lösungsspaziergang zu den Aufgaben für heute.
- **Heimarbeit bis Freitag, den 26.11., 17:00.** Gib deine Lösungen zu der ⭐-Aufgabe von diesem Übungsblatt in Moodle ab. (Feste Abgabefrist!)

Dienstag

Aufgabe 6.1 (Erwartungswerte).

a) Sei X eine Zufallsvariable, die die Werte 2, 5 und 8 mit den folgenden Wahrscheinlichkeiten annimmt:

$$\Pr[X = 2] = \tfrac{1}{3},$$
$$\Pr[X = 5] = \tfrac{1}{2},$$
$$\Pr[X = 8] = \tfrac{1}{6}.$$

Was ist der Erwartungswert von X?

b) Sei Y eine Zufallsvariable, die den Wert 2^i für alle $i \in \{0, 1, 2, \dots\}$ mit Wahrscheinlichkeit $\Pr[Y = 2^i]$ annimmt, wobei

$$\Pr[Y = 2^i] = \frac{1}{2^{(i+1)}}\,.$$

Was ist der Erwartungswert von Y?

Aufgabe 6.2 (Analyse von Selection). In Abschnitt 13.5 (**KT**) wurde die erwartete Laufzeit von Selection analysiert. Anstatt eine Phase wie im Buch zu definieren, definieren wir eine Phase wie folgt: Der Algorithmus ist in Phase j, wenn die Größe g der betrachteten Menge im folgenden Intervall liegt

$$n\left(\frac{2}{3}\right)^{j+1} < g \leq n\left(\frac{2}{3}\right)^{j}.$$

Führe die Analyse von Selection mit der neuen Phasendefinition durch. Erreicht man mit dieser Analyse im Erwartungswert immer noch eine lineare Laufzeit?

Aufgabe 6.3 (Weihnachtsfeier im Institut). Während der Weihnachtsfeier des Instituts für Informatik möchte Professor Regloh wissen, wer die meisten Cookies im „Bing or Ding"- Spiel gewonnen hat. Er schlägt den folgenden Algorithmus vor:

Algorithm 1: Finde Student:in mit den meisten Cookies

---

max $\leftarrow -\infty$
$s \leftarrow$ null
Permutiere die Student:innen gleichverteilt zufällig.
Sei $s_1, \dots, s_n$ die gezogene Reihenfolge.
Sei c_i die von Student:in s_i gewonnene Anzahl an Cookies.
for $i = 1, \dots, n$ **do**
    if $c_i >$ max **then**
        max $\leftarrow c_i$ and $s \leftarrow s_i$ $(*)$
    end
end
return s

Nimm an, dass alle Student:innen eine unterschiedliche Anzahl an Cookies gewonnen haben, d.h. $c_i \neq c_j$ gilt für alle i, j mit $i \neq j$.

a) Was ist die Wahrscheinlichkeit, dass die im Code mit $(*)$ markierte Zeile in der letzten Iteration ausgeführt wird?
b) Sei X_i eine Zufallsvariable. Sie nimmt den Wert 1 an, wenn die $(*)$-Zeile in Iteration i ausgeführt wird, und 0 wenn nicht. Berechne $\Pr(X_i = 1)$.
c) Wie oft wird die $(*)$-Zeile in Erwartung ausgeführt? Berechne den Erwartungswert.

Donnerstag

Aufgabe 6.4 (Schrauben und Muttern). Beim Ausmisten deines Kellers hast du genau N Muttern und N Schrauben gefunden. Intuitiv weißt du, dass jede Mutter zu *genau* einer Schraube und jede Schraube zu *genau* einer Mutter passt, obwohl es nicht möglich ist, ein Paar von Mutter und Schraube mit bloßem Auge zu vergleichen. Um einen Vergleich durchzuführen, musst du die Mutter an der Schraube anbringen und herauszufinden, welches der beiden Teile größer ist.

Für dieses Problem ist nur ein sehr komplizierter deterministischer Algorithmus mit Laufzeit $\mathcal{O}(N \log N)$ bekannt. Gib einen effizienten randomisierten Algorithmus an, der so schnell wie möglich herausfindet, welche Paare zusammenpassen.

Hinweis: Modifiziere QuickSort für dieses Problem.

Aufgabe 6.5 (Bierkisten). Von n Bierkisten $B_1, \dots, B_n$ enthalten genau k Kisten eine Flasche Bier ($k \leq n$), während die restlichen Kisten leer sind. Du kannst mit bloßem Auge nicht sehen, welche der Kisten leer sind und welche nicht. Das Ziel ist es, eine Flasche Bier in einer Kiste zu finden. Dafür schlagen wir einen deterministischen und zwei randomisierte Algorithmen vor:

> FINDEBIER:
> Öffne die Kisten $B_1, B_2, \dots$ der Reihenfolge nach, wobei das Öffnen einer Kiste einem Rechenschritt entspricht. Der Algorithmus terminiert, wenn ein Bier gefunden wurde.
>
> FINDEZUFÄLLIGBIER:
> Wähle zufällig eine Zahl $i \in \{1, 2, \dots, n\}$ und öffne die Kiste B_i. Der Algorithmus terminiert, wenn ein Bier gefunden wurde.
>
> FINDEZUFÄLLIGUNGEÖFFNETESBIER:
> Wähle zufällig eine noch nicht geöffnete Kiste aus und öffne diese. Der Algorithmus terminiert, wenn ein Bier gefunden wurde.

a) Was ist die *best-case* Laufzeit von FINDEBIER?
b) Was ist die *worst-case* Laufzeit von FINDEBIER?
c) Was ist die erwartete Laufzeit von FINDEZUFÄLLIGBIER?
d) Was ist die *worst-case* Laufzeit von FINDEZUFÄLLIGBIER?
e) Was ist die *worst-case* Laufzeit von FINDEZUFÄLLIGUNGEÖFFNETESBIER?
f) Was ist die erwartete Laufzeit von FINDEZUFÄLLIGUNGEÖFFNETESBIER?

   Betrachte hierzu folgende Hilfestellung: Sei X die Anzahl der vom Algorithmus geöffneten Kisten. Sei X_i eine Indikatorvariable für jede *leere* Kiste i. Wenn die Kiste geöffnet wurde, ist $X_i = 1$, sonst ist $X_i = 0$.

   f_1) Stelle X durch die X_i dar, in der Form $X = \dots\dots$
   f_2) Was ist der Erwartungswert von X_i?
   f_3) Benutze die Erwartungswerte der X_i, um den Erwartungswert von X anzugeben.
   f_4) Was ist die erwartete Laufzeit von FINDEZUFÄLLIGUNGEÖFFNETESBIER?

Sternaufgabe

Aufgabe 6.6 (⭐: Randomisierte Pivotwahl). Die Laufzeit von *Quicksort* und *Quickselect* hängt ganz entscheidend von der Wahl des Pivotelements ab. Das Pivotelement sollte das zu sortierende Array in zwei möglichst gleichgroße Teilarrays aufspalten.

Gegeben sei ein unsortiertes Array $A[1..n]$ mit n paarweise verschiedenen Elementen $A[1], \dots, A[n]$. Wir nehmen an, dass n ganzzahlig durch 4 teilbar ist. Weiterhin sei der *Rang* $r(x)$ *von* x die Position des Elements x im sortierten Array. Zum Beispiel heißt $r(x) = 3$, dass x das drittkleinste Element von A ist.

Eine mögliche Strategie für die Pivotwahl ist Folgende:

> $\textsc{SamplePivot}(A)$:
>
> - Wähle uniform zufällig 7 Elemente $p_1, p_2, \dots, p_7$ aus dem Array A aus. Hierbei können Elemente mehrmals gewählt werden („Ziehen mit Zurücklegen").
> - Liefere den Median dieser 7 Elemente zurück.

a) Was ist die Laufzeit von $\textsc{SamplePivot}$? (Ohne Begründung.)

b) Berechne $\Pr\left(\frac{n}{4} < r(p_i) \leq \frac{3n}{4}\right)$ für alle $i \in \{1, \dots, 7\}$ und begründe deine Antwort.

c) Sei
$$X_i = \begin{cases} 1, & \text{wenn } \frac{n}{4} < r(p_i) \leq \frac{3n}{4}; \\ 0, & \text{sonst.} \end{cases}$$
Sei $X = \sum_{i=1}^{7} X_i$. Berechne $\Pr(X \geq 4)$ und begründe deine Antwort.

d) Berechne $\Pr\left(\frac{n}{4} < r(\textsc{SamplePivot}(A)) \leq \frac{3n}{4}\right)$ und begründe deine Antwort.

e) Wir wiederholen $\textsc{SamplePivot}(A)$ jetzt unabhängig so oft, bis das Ereignis
$$\frac{n}{4} < r(\textsc{SamplePivot}(A)) \leq \frac{3n}{4}$$
zum ersten Mal eintritt. Sei W die Anzahl der Wiederholungen. Was ist der Erwartungswert von W? Begründe deine Antwort.