Miniquiz

Wenn ein probabilistischer Algorithmus mit einer Wahrscheinlichkeit von mindestens $2/3$ eine korrekte JA/NEIN-Antwort liefert und wir ihn unabhängig n-mal ausführen, ist die Wahrscheinlichkeit, dass er mehr als $n/2$ Mal eine falsche Antwort gibt, kleiner als $\exp(-0.001n)$.

If a probabilistic algorithm provides a correct YES/NO answer with probability at least $2/3$, and we run it independently n times, probability that it gives the wrong answer more than $n/2$ times is smaller than $\exp(-0.001n)$.

Chernoff: $\Pr[X > \frac{n}{2}] \leq \Pr[X \geq \frac{n}{2}] = \Pr[X \geq (1+\frac{1}{6})\frac{n}{3}] \leq \Pr[X \geq (1+\frac{1}{6})E[X]]$ $\leq e^{-\frac{1}{6^2}\cdot\frac{1}{3}\cdot\frac{n}{3}} \leq e^{-\frac{n}{1000}}$

(X: #falschen Antworten)

- [x] Wahr
- [ ] Falsch

Es gibt einen probabilistischen Algorithmus, der testet, ob n eine Primzahl ist, und zwar in einer Zeit, die polynomiell in $\log n$ ist.

There is a probabilistic algorithm testing if n is a prime number in time polynomial in $\log n$.

- [x] Wahr
- [ ] Falsch

Sei $\mathcal{A}$ ein probabilistischer Algorithmus, dessen Ausgabe immer entweder 0 oder 1 ist, und gelte

$$\mathbb{E}[\mathcal{A}] = s > 0.$$

Seien außerdem $0 < \varepsilon < 1$ und $0 < \delta < 1$. Wenn wir $\mathcal{A}$ unabhängig

$$m = \left\lceil \frac{100 \log(1/\delta)}{s\varepsilon^2} \right\rceil$$

($m \in \mathbb{N}$) Mal ausführen und $\hat{s}$ als Durchschnitt der Ausgaben definieren, dann gilt mit Wahrscheinlichkeit mindestens $1 - \delta$:

$$\hat{s} \in [(1-\varepsilon)s, (1+\varepsilon)s].$$

- **Target Shooting:** If the target-shooting algorithm identifies a set $S \subseteq U$ with $N \geq 3\frac{|U|}{|S|}\varepsilon^{-2}\ln(2/\delta)$ trials, then with probability $\geq 1-\delta$ the output lies in the interval

($|S| \leq |U|$)

$$\left[(1-\varepsilon)\frac{|S|}{|U|}, (1+\varepsilon)\frac{|S|}{|U|}\right].$$

(#1 ≤ Output ≤ 1)

times and define $\hat{s}$ as the average of the outputs, then with probability at least $1-\delta$,

$$\hat{s} \in [(1-\varepsilon)s, (1+\varepsilon)s].$$

$100 \cdot \log(\frac{1}{\delta}) \cdot \frac{1}{s} \cdot \varepsilon^{-2} \geq 3\frac{1}{s}\varepsilon^{-2} \cdot \log(\frac{2}{\delta})$

- [x] Wahr
- [ ] Falsch

Wir können jeden Las-Vegas-Algorithmus mit bekannter erwarteter Laufzeit höchstens T in einen randomisierten Algorithmus umwandeln, dessen Laufzeit höchstens $10T$ ist und der mit Wahrscheinlichkeit mindestens 0.9 erfolgreich ist.

We can transform every Las Vegas algorithm with known expected running time at most T into a randomized algorithm whose running time is at most $10T$ and whose success probability is at least 0.9.

$T' \geq 0$

Markov: $\Pr[T' \geq 10T] \leq \frac{T}{10T} = \frac{1}{10}$

$E[\underbrace{\text{Laufzeit}}_{T'}] \leq T$

- [x] Wahr
- [ ] Falsch

Jeder Monte Carlo Algorithmus kann in einen Las Vegas Algorithmus umgewandelt werden.

Every Monte Carlo algorithm can be converted into a Las Vegas algorithm.

- ○ Wahr
- ⊗ Falsch

Sarah besitzt zwei sehr spezielle Münzen: eine zeigt immer 'Kopf' (weil auf beiden Seiten 'Kopf' abgebildet ist ...), die andere immer 'Zahl'. Sie wählt eine der beiden Münzen zufällig und wirft sie $n = 1000$ mal. Sie bezeichnet mit X die Anzahl 'Kopf' die sie sieht.

Dann gilt für $\delta := 1/4$:

$\Pr[X \geq (1+\delta)n/2] \leq e^{-\delta^2 n/6}$.

Sarah has two special coins: one always lands heads, and the other always lands tails. She picks one of the two coins uniformly at random and flips it $n = 1000$ times. Let X be the number of heads observed.

Then, for $\delta := 1/4$,

$$\Pr\left[X \geq (1+\delta)\frac{n}{2}\right] \leq e^{-\delta^2 n/6}.$$

- ○ Wahr
- ⊗ Falsch

Ein deterministischer Algorithmus kann immer als randomisierter Algorithmus gesehen werden.

A deterministic algorithm can always be viewed as a randomized algorithm.

- ⊗ Wahr
- ○ Falsch

Seien X, Y, Z drei Zufallsvariabeln wobei X, Y, Z unabhängig sind, dann gilt immer

$$E[X + Y \cdot Z] = E[X] + E[Y] \cdot E[Z].$$

Let X, Y, Z be three random variables such that X, Y, Z are independent. Then it always holds that

$$E[X + Y \cdot Z] = E[X] + E[Y] \cdot E[Z].$$

- ⊗ Wahr
- ○ Falsch

Die erwartete Laufzeit von Quickselect ist $O(n)$.

The expected running time of Quickselect is $O(n)$.

- ⊗ Wahr
- ○ Falsch

Quicksort ist ein Monte-Carlo-Algorithmus.

Quicksort is a Monte Carlo algorithm.

- ○ Wahr
- ⊗ Falsch

Besprechung: PG04

Exercise P4.1 – *Randomized Algorithms*

We want to solve the following problem: Given a graph $G = (V, E)$ and a natural number k, find a subset of edges $E' \subseteq E$ such that $G' = (V, E')$ is k-colorable. Describe a randomized algorithm that computes such a set E' with $|E'| > \frac{k-2}{k}|E|$. The algorithm should have linear expected runtime. That is, if X denotes the random variable that counts the number of elementary operations executed by the algorithm, then $\mathbb{E}[X]$ should be $\Theta(|V| + |E|)$ for every input.

Hint: You could proceed as follows: (i) find a linear-time algorithm that has a positive constant probability of finding such a set E'; (ii) repeat this algorithm until it is successful; (iii) bound the expected number of repetitions needed.

$\mathcal{A}$: färbe jeden Knoten mit W'keit $\frac{1}{k}$ mit i-ter Farbe (insgesamt k Farben)
→ löschen von $e = \{u, v\}$, $c(u) = c(v)$

$O(|V| + |E|)$

$X_e = \begin{cases} 1 & c(u) = c(v) \\ 0 & \end{cases}$

$E' = E \setminus E''$

$$\mathbb{E}[|E''|] = \mathbb{E}\left[\sum_{e \in E} X_e\right] = \sum_{e \in E} \underbrace{\mathbb{E}[X_e]}_{= \frac{1}{k}} = |E| \cdot \frac{1}{k}$$

Markov: $E'' \geq 0$

$$\Pr\left[|E''| \geq \frac{2}{k}|E|\right] \leq \frac{\mathbb{E}[|E''|]}{\frac{2}{k}|E|} = \frac{1}{2}$$

$\mathcal{A}'$: Wdh. $\mathcal{A}$ solange bis Erfolg
→ geom. verteilt mit $p = \frac{1}{2}$
→ zweimal ausführen im Erwartungswert
→ erwartete Laufzeit: $O(|V|+|E|)$ ✓

HS17 Aufgabe 2c

4/80 Punkten ⇒ ca. 9 Minuten Zeit

(c) Beim Miller-Rabin-Primzahltest nutzt man einen Test $T : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ (eine Subroutine) mit den folgenden beiden Eigenschaften aus:

- $T(a, p) = 1$, falls $p > 2$ eine Primzahl ist und $a \in \{1, \ldots, p-1\}$.
- $\Pr[T(a, p) = 1] \leq 1/4$, falls $p > 2$ *keine* Primzahl ist und a uniform zufällig aus $\{1, \ldots, p-1\}$ ist.

Geben Sie einen möglichst effizienten Monte-Carlo-Algorithmus an, der entscheidet, ob ein $p > 2$ eine Primzahl ist, und dessen Fehlerwahrscheinlichkeit höchstens 10^{-4} ist. (Sie brauchen *nicht* darauf eingehen, wie der Test T aussieht, oder wie er implementiert wird.)

(4 Punkte)

Fehlerreduktionen:

- **Wiederholung MC:** Eine N-fache Wiederholung mit $N = 4\varepsilon^{-2} \ln \delta^{-1}$ steigert die Erfolgswahrscheinlichkeit eines Monte-Carlo-Algorithmus von $\frac{1}{2} + \varepsilon$ auf $\geq 1 - \delta$.
- **Wiederholung MC mit einseitigem Fehler:** Eine N-fache Wiederholung mit $N = \varepsilon^{-1} \ln \delta^{-1}$ steigert fÃ¼r einen Monte-Carlo-Algorithmus mit einseitigem Fehler die Erfolgswahrscheinlichkeit von ε auf $\geq 1 - \delta$.
- **Target Shooting:** Bestimmt der Target-Shooting-Algorithmus eine Menge $S \subseteq U$ mit $N \geq 3\frac{|U|}{|S|}\varepsilon^{-2} \ln(2/\delta)$ Versuchen, so ist die Ausgabe mit Wahrscheinlichkeit $\geq 1 - \delta$ im Intervall $\left[(1-\varepsilon)\frac{|S|}{|U|}, (1+\varepsilon)\frac{|S|}{|U|}\right]$.

```
N := 16
for i=1..N do
   wähle a∈{1,...,p-1} uniform zufällig
   if T(a,p)=0 then return "p keine Primzahl"
return "Primzahl"
```

$$N = \frac{1}{\varepsilon} \cdot \ln\left(\frac{1}{\delta}\right) = \frac{4}{3} \cdot \ln(10^4) = \frac{16}{3} \ln(10) \leq 16$$

Hashing

Hashfunktion $h: U \to [m]$ ($[m] = \{1,\dots,m\}$), gegeben Datensatz $S = \{s_1,\dots,s_n\}$

→ effizient berechenbar

→ schwer umzukehren

→ $\forall i \in [m] \quad Pr[h(s) = i] = \frac{1}{m}$

$s_i = s_j \not\Longleftarrow \Rightarrow h(s_i) = h(s_j)$

$\#\text{Kollisionen} = \#\text{Duplikate} + \underbrace{\#\text{unbeabsichtigte Kollisionen}}_{x}$

Indikatorvariable X_{ij}: $X_{ij} = 1 \iff$ es gibt Kollision mit Daten s_i und s_j, $s_i \neq s_j$

$$Pr[X_{ij} = 1] = \begin{cases} \frac{1}{m} & \text{falls } s_i \neq s_j \\ 0 & \text{sonst} \end{cases}$$

$E[X_{ij}] \leq \frac{1}{m}$

$E[X] = \sum_{1 \leq i < j \leq n} E[X_{ij}] \leq \binom{n}{2} \cdot \frac{1}{m}$

QUICKSELECT(A, ℓ, r, k)

```
1: p ← Uniform({ℓ, ℓ+1, ..., r})          ▷ wähle Pivotelement zufällig
2: t ← PARTITION(A, ℓ, r, p)
3: if t = ℓ + k − 1 then
4:     return A[t]                          ▷ gesuchtes Element ist gefunden
5: else if t > ℓ + k − 1 then
6:     return QUICKSELECT(A, ℓ, t − 1, k)   ▷ gesuchtes Element ist links
7: else
8:     return QUICKSELECT(A, t + 1, r, k − t)  ▷ gesuchtes Element ist rechts
```

zufällige Folge der Form $(\ell_0, r_0, k_0), (\ell_1, r_1, k_1), \ldots, (\ell_N, r_N, k_N)$
mit $(\ell_0, r_0, k_0) = (1, n, k)$

$$(\ell_{i+1}, r_{i+1}) = \begin{cases} (\ell_i, t-1) \\ (t+1, r_i) \end{cases}$$

$T :=$ #gemessene Vergleiche von Elementen

$T =$

$N_j :=$ #Aufrufe von Quickselect für die gilt $\left(\frac{3}{4}\right)^j < r_i - \ell_i + 1 \leq \left(\frac{3}{4}\right)^{j-1} n$

Dann gilt:

Recap: Multiple Choice

1) Sei G ein Graph, der einen Hamiltonkreis hat.
Dann ist G 2-zusammenhängend.
→ Wahr

2) Ein Graph kann niemals 5-kantenzusammenhängend sein, wenn er...
... Knoten vom Grad 3 hat. → Wahr
... nicht 5-zusammenhängend ist. → Falsch
... nicht 4-zusammenhängend ist. → Falsch

Knotenzusammenhang
≤ Kantenzusammenhang
≤ Minimalgrad

3) Wie viele Artikulationsknoten hat ein Pfad der Länge 5?
→ 4

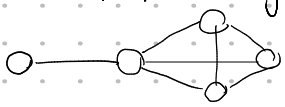
4) Sei G ein Graph mit $n \geq 3$ und sei $\deg(v) \geq \frac{1}{2}n$ für alle $v \in V$.
Dann ist G zusammenhängend.
→ Wahr, Satz von Dirac

5) In jedem Graphen, welcher eine gerade Knotenanzahl und einen Hamiltonkreis hat, gibt es ein perfektes Matching.
→ Wahr

6) Sei M ein Matching in einem Graphen G. Dann gibt für jeden M-augmentierenden Pfad P, dass er Länge mind. $\frac{|M|}{2}$ hat.
→ Falsch

7) Wenn ein Matching inklusionsmaximal ist, gibt es keine augmentierenden Pfade mehr
→ Falsch

8) Sei G ein Graph. Wenn es einen Knoten $v \in V(G)$ gibt, für den gilt, dass $G[N(v)]$ mit k Farben gefärbt werden kann, dann kann G mit k+1 Farben gefärbt werden.
→ Falsch



9) Der Greedy-Algorithmus findet immer eine optimale Färbung, wenn er die Knoten in der richtigen Reihenfolge durchläuft.
→ Richtig

10) Seien A, B, C unabhängige Ereignisse mit $Pr[A \cap B \cap C] > 0$.

- $Pr[(A \cup B) \cap C] = (Pr[A] + Pr[B]) \cdot Pr[C]$, Falsch $A = B = \Omega$
- $Pr[A] + Pr[B] \leq Pr[A \cup B]$, Falsch, $Pr[A \cup B] = Pr[A] + P[B] - Pr[A \cap B]$

11) Wir können jeden Las Vegas Algorithmus mit erwarteter Laufzeit T in einen randomisierten Algorithmus mit Laufzeit höchstens 10T und Erfolgsw'keit mind. 0,9 umwandeln.
→ s. Miniquiz

12) Wir wissen, dass $ggT(9973, 100!) = 1$ gilt. Folgt daraus, dass 9973 prim ist?
→ Wahr

13) Betrachte den Miller-Rabin-Primzahltest

(a) Die Ausgabe "keine Primzahl" ist immer richtig

(b) Die Ausgabe "Primzahl" ist immer richtig

14) Wir nehmen an, dass $A, B \subseteq \Omega$ unabhängig sind.
Sind auch A, B und Ω unabhängig?
$Pr[A \cap \Omega] = Pr[A] = Pr[\Omega] \cdot Pr[A]$ (analog für B)
$Pr[A \cap B \cap \Omega] = Pr[A \cap B] = Pr[A] \cdot Pr[B] \cdot \underbrace{Pr[\Omega]}_{=1}$

15) Seien $X_1 \sim Bin(n,p)$ und $X_2 \sim Bin(n,p)$ zwei unabhängige binomelle ZV.
Welcher Verteilung folgt $X_1 + X_2$?
→ $X_1 + X_2 \sim Bin(2n,p)$

16) Seien X, Y, Z ZV wobei X, Y unabhängig sind, dann gilt immer
$E[X + Y \cdot Z] = E[X] + E[Y] \cdot E[Z]$
→ Falsch

17) Seien X und Y zwei ZV mit $E[XY] = E[X]E[Y]$. Dann sind X und Y unabhängig
→ Falsch, andere Implikation stimmt
$X(\omega) = \omega \quad \Omega = \{-1, 0, 1\} \quad Y = \begin{cases} 1 & \text{falls } X = 0 \\ 0 & \text{sonst} \end{cases}$

18) Der randomisierte Algorithmus A löst Problem P mit W'keit p und gibt sonst "keine Antwort" aus.
Wie oft müssen wir A im Erwartungswert laufen lassen, bis er Problem P löst?
→ geometrisch verteilt, also $\frac{1}{p}$