Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Peter Sanders
Dr. Thomas Worsch, Dr. Simon Gog
Demian Hespe, Yaroslav Akhremstev

1. Übungsblatt zu Algorithmen II im WS 2017/2018

http://algo2.iti.kit.edu/AlgorithmenII_WS17.php
{hespe,sanders,simon.gog,worsch,yaroslav.akhremtsev}@kit.edu

Musterlösungen

Aufgabe 1 (*Schlechter Zufall*)

Gegeben sei ein randomisierter Algorithmus badBit, der keine Eingabe liest und als Ausgabe zufällig mit Wahrscheinlichkeit p die Zahl 0 und mit Wahrscheinlichkeit $q = 1 - p$ die Zahl 1 liefert. Es sei $0 < p < 1$; der konkrete Wert von p sei aber unbekannt.
Entwerfen Sie einen randomisierten Algorithmus fairBit, der keine Eingabe liest und als Ausgabe immer zufällig eine der Zahlen 0 und 1 mit Wahrscheinlichkeit 1/2 liefert.
Was können Sie über die Laufzeit Ihres Algorithmus sagen?

Musterlösung:
Algorithmusidee:

```
1: function fairBit
2:     repeat
3:         x ← badBit()
4:         y ← badBit()
5:     until x ≠ y
6:     return x
7: end function
```

Korrektheit:
$\mathbb{P}(x = 1 | x \neq y) = \frac{\mathbb{P}(x=1 \wedge y=0)}{\mathbb{P}(x \neq y)} = \frac{pq}{2pq} = \frac{1}{2}$

Laufzeit:
Im Folgenden benutzen wir folgende Überlegung (für $z \in \mathbb{R}$ mit $0 < |z| < 1$):

$$\text{sei} \qquad R = \sum_{i=1}^{\infty} i \cdot z^{i-1} = \sum_{i=0}^{\infty} (i+1) \cdot z^i$$

$$\text{dann} \qquad Rz = \sum_{i=1}^{\infty} i \cdot z^i$$

$$\text{also} \qquad R(1-z) = R - Rz = \sum_{i=0}^{\infty} z^i = \frac{1}{1-z}$$

$$\text{also} \qquad R = \frac{1}{(1-z)^2}$$

Nun ist im Algorithmus die Wahrscheinlichkeit für $x = y$ gleich $z = p^2 + q^2$, also $1 - z = 2pq$. Wegen $0 < p < 1$ ist $0 < |z| < 1$. Daher ergibt sich für den Erwartungswert der Laufzeit:

$$\sum_{i=1}^{\infty} i \cdot 2 \cdot (p^2 + q^2)^{i-1} 2pq = 4pq \sum_{i=1}^{\infty} i \cdot z^{i-1} = 4pq \frac{1}{4p^2q^2} = \frac{1}{pq}$$

Aufgabe 2 (*Zufällige Primzahlen*)

Gegeben sei ein deterministischer Algorithmus isPrime, der eine natürliche Zahl n als Eingabe, überprüft ob sie eine Primzahl ist oder nicht.
Finden Sie einen randomisierten Algorithmus, der zu einer natürlichen Zahl $m \geq 2$ als Eingabe als Ausgabe zufällig gleichverteilt jede Primzahl p mit $2 \leq p \leq m$ aus Ausgabe produziert.
Was können Sie über die Laufzeit Ihres Algorithmus sagen?

Musterlösung:
Algorithmusidee:

```
1: function randPrime(m)
2:     repeat
3:         k ← randInteger(2..m)
4:     until isPrime(k)
5:     return k
6: end function
```

Im Bereich $2..m$ gibt es näherungsweise $\pi(m) = m/\ln m$ viele Primzahlen. Also Erwartungswert für die Anzahl Aufrufe von randPrime:

$$\sum_{i=1}^{\infty} i \left(1 - \frac{1}{\ln m}\right)^{i-1} \frac{1}{\ln m} = \frac{1}{\ln m} \cdot \frac{1}{1/(\ln m)^2} = \ln m$$

Aufgabe 3 (*Cover Time*)

Für einen Knoten $v \in V$ eines ungerichteten zusammenhängen Graphen $G = (V, E)$ ist die *Überdeckungszeit* C_v der Erwartungswert für die Anzahl Schritte, die ein Random Walk, der in v startet, benötigt, bis zum ersten Mal jeder Knoten mindestens einmal besucht worden ist.
Zeigen Sie, dass für den vollständigen Graphen K_n mit n Knoten die Überdeckungszeit $C(n) \in O(n \log n)$ ist. (Hier ist der Anfangsknoten offensichtlich irrelevant.) Bearbeiten Sie dazu die folgenden Punkte:

- Angenommen, zu einem Zeitpunkt sind bereits i Knoten besucht worden. Wie groß ist dann die Wahrscheinlichkeit, dass im nächsten Schritt ein „neuer“ Knoten besucht wird?
- Es sei X_i die Zufallsvariable, die angibt, wieviele Schritte der Random Walk braucht von dem Zeitpunkt, zu dem zum ersten Mal i verschiedene Knoten besucht sind, bis zu dem Zeitpunkt, zu dem zum ersten Mal $i + 1$ verschiedene Knoten besucht sind. Berechnen Sie $\mathbb{E}[X_i]$.
- Wie kann man mit Hilfe der X_i die gesuchte Größe C ausdrücken?

Musterlösung:
Es sei $n = |V|$.

- Da jeder der $n - 1$ Knoten mit gleicher Wahrscheinlichkeit als nächster zu besuchender gewählt wird, und von diesen $i - 1$ schon mal besucht wurden (ein besuchter ist der aktuelle), ist die gesuchte Wahrscheinlichkeit
$$p_i = \frac{n - 1 - (i - 1)}{n - 1} = \frac{n - i}{n - 1}$$
- Es ist
$$\mathbb{E}[X_i] = \frac{1}{p_i} = \frac{n - 1}{n - i}$$
- Es ist
$$C = \mathbb{E}[\sum_{i=1}^{n-1} X_i] = \sum_{i=1}^{n-1} \frac{n - 1}{n - i} = (n - 1) \sum_{i=1}^{n-1} \frac{1}{n - i} = (n - 1) H_{n-1} \in \Theta(n \log n)$$

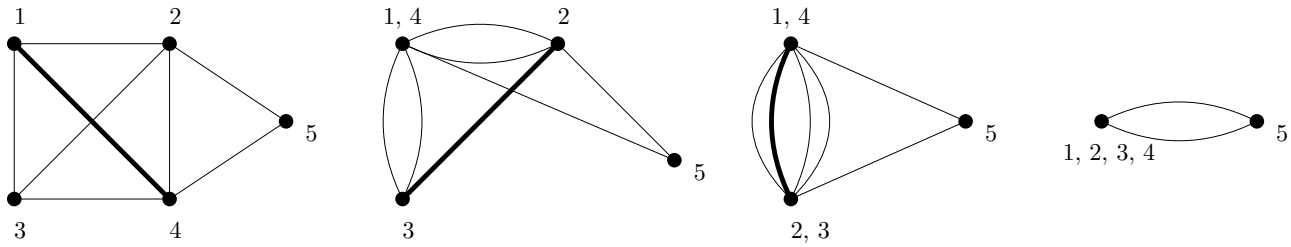

Abbildung 1: Kontraktion von Kanten bis ein Schnitt der Größe 2 gefunden wurde mit $S = \{5\}$

Aufgabe 4 (*Kugeln ziehen*)

Gegeben sei folgendes Spiel: In einer Urne befinden sich eine weiße und eine schwarze Kugel. In jeder Runde ziehen Sie zufällig gleichverteilt eine Kugel aus der Urne, legen sie zurück und legen eine weitere Kugel der gleichen Farbe hinzu.
Zeigen Sie, dass nach n Runden, die Wahrscheinlichkeiten $1..n-1$ weiße Kugeln in der Urne zu haben, gleich sind.

Musterlösung:
Beweis per Induktion:

- **Induktionsanfang**: In der ersten Runde ist die Wahrscheinlichkeit eine weiße Kugel zu ziehen 0.5. Nach der ersten Runde befinden sich also mit Wahrscheinlichkeiten jeweils 0.5 eine oder zwei weiße Kugeln in der Urne.
- **Induktionsvoraussetzung**: Nach n Runden beträgt die Wahrscheinlichkeit i weiße Kugeln in der Urne zu haben $\frac{1}{n-1}$ für alle $i \in \{1..n-1\}$.
- **Induktionsschritt**: $n \to n+1$
  Sei X_n die Anzahl der weißen Kugeln in der Urne nach n Runden.
  $\mathbb{P}(X_{n+1} = i) = \mathbb{P}(X_n = i-1) \cdot \frac{i-1}{n} + \mathbb{P}(X_n = i) \cdot \frac{n-i}{n}$
  $= \begin{cases} \frac{1}{n-1} \cdot \frac{i-1}{n} + \frac{1}{n-1} \cdot \frac{n-i}{n} = \frac{1}{n} & \text{falls } 1 < i < n \\ 0 + \frac{1}{n-1} \cdot \frac{n-1}{n} = \frac{1}{n} & \text{falls } i = 1 \\ \frac{1}{n-1} \cdot \frac{n-1}{n} + 0 = \frac{1}{n} & \text{falls } i = n \end{cases}$ *Nach Induktionsvoraussetzung*

Aufgabe 5 (*Min Cut*)

Gegeben sei ein ungerichteter Graph $G = (V, E)$, finden Sie S $(S \neq \emptyset \wedge S \neq V)$, so dass $|C|$ mit $C = \{\{u, v\} \in E : u \in S \wedge v \in V - S\}$ minimal wird. Gesucht wird also eine Aufteilung des Graphen in zwei nichtleere Mengen, so dass die Anzahl der Kanten, die zwischen beiden Mengen verlaufen, minimiert wird. Diese Menge C nennt man einen minimalen Schnitt.
Wir verwenden folgenden randomisierten Algorithmus zur Lösung dieses Problems: In jedem Schritt wählen wir zufällig gleichverteilt eine Kante $e = \{u, v\}$ und *kontrahieren* die beiden inzidenten Knoten zu einem neuen Knoten w, indem wir u, v und e aus dem Graphen entfernen und in allen verbleibenden Kanten, die u oder v als Endpunkt haben, w einsetzen. Dabei können parallele Kanten entstehen, die wir im Graphen belassen! Der Algorithmus endet, wenn nur noch zwei Knoten verbleiben. Die Knoten, aus denen die verbleibenden Knoten entstanden sind, bilden die beiden gesuchten Mengen. Abbildung**??** soll den Algorithmus veranschaulichen.

- Zeigen Sie, dass dieser Algorithmus mit einer Wahrscheinlichkeit $\geq \frac{2}{n(n-1)}$ einen minimalen Schnitt findet.
  Zur Hilfe können Sie sich an folgendem Schema orientieren, in dem Sie einen minimalen Schnitt C mit $|C| = k$ betrachten und die Wahrscheinlichkeit berechnen, dass keine der Kanten in C kontrahiert werden.

- Betrachten Sie ein Ereignis E_i, dass angibt, dass die Kante, die in Iteration i kontrahiert wird, nicht in C enthalten ist. Konstruieren sie daraus das Ereignis F_i, das angibt, dass keine der Kanten aus C in den ersten i Iterationen kontrahiert wurde.
- Als ersten Schritt berechnen Sie $\mathbb{P}(E_1)$ und $\mathbb{P}(F_1)$ (überlegen Sie sich, was der minimale Knotengrad in G ist).
- Berechnen Sie dann $\mathbb{P}(E_i|F_{i-1})$.
- Berechnen Sie $\mathbb{P}(F_i)$ für ein geeignetes i (Hinweis: Schreiben Sie sich einzelne Produktglieder auf und kürzen Sie).

- Der Algorithmus wird nun $n(n-1)\ln n$ mal ausgeführt und am Ende der kleinste gefundene Schnitt zurückgegeben. Zeigen Sie, dass die Wahrscheinlichkeit, dass die Ausgabe kein minimaler Schnitt ist, kleiner als $\frac{1}{n^2}$ ist. (Hinweis: $1 - x \leq e^{-x}$)

Musterlösung:
Es gilt $F_i = \bigcap_{j=1}^{i} E_j$
Der minimale Knotengrad in G ist mindestens k. Andernfalls wäre mit $S = \{v\}$ für einen Knoten mit $\text{Grad}(v) < k$ ein Schnitt der Größe $\text{Grad}(v) < k$ gefunden. Es gibt also mindestens $nk/2$ Kanten. Daher,

$$\mathbb{P}(E_1) = \mathbb{P}(F_1) \geq 1 - \frac{2k}{nk} = 1 - \frac{2}{n}$$

Nach einer Iteration gilt dann

$$\mathbb{P}(E_2|F_1) \geq 1 - \frac{k}{k(n-1)/2} = 1 - \frac{2}{n-1},$$

da jetzt ein Knoten weniger vorhanden ist.

Ähnlich gilt:

$$\mathbb{P}(E_i|F_{i-1}) \geq 1 - \frac{k}{k(n-i+1)/2} = 1 - \frac{2}{n-i+1}$$

Nach $n-2$ Iterationen sind nur noch 2 Knoten übrig. Dann gilt

$$\begin{aligned}
\mathbb{P}(F_{n-2}) &= \mathbb{P}(E_{n-2} \cap F_{n-3}) = \mathbb{P}(E_{n-2}|F_{n-3}) \cdot \mathbb{P}(F_{n-3}) \\
&= \mathbb{P}(E_{n-2}|F_{n-3}) \cdot \mathbb{P}(E_{n-3}|F_{n-4}) \cdot \cdots \cdot \mathbb{P}(E_2|F_1) \cdot \mathbb{P}(F_1) \\
&\geq \prod_{i=1}^{n-2}(1 - \frac{2}{n-i+1}) = \prod_{i=1}^{n-2}(\frac{n-i-1}{n-i+1}) \\
&= (\frac{n-2}{n})(\frac{n-3}{n-1})(\frac{n-4}{n-2})\ldots(\frac{4}{6})(\frac{3}{5})(\frac{2}{4})(\frac{1}{3}) = \frac{2}{n(n-1)}
\end{aligned}$$

Nach $n(n-1)\ln n$ Iterationen erhalten wir

$$(1 - \frac{2}{n(n-1)})^{n(n-1)\ln n} \leq e^{-2\ln n} = \frac{1}{n^2}$$