Karlsruher Institut für Technologie

Maximilian Katzmann
Hans-Peter Lehmann
Stefan Walzer

Übungsblatt 14 – Aktivsession

Randomisierte Algorithmik – Wintersemester 2023/2024

Folgende Aufgaben werden in der Aktiv-Session am 8.2.2024 gemeinsam bearbeitet. Gib eine Lösung zu zwei *der folgenden Aufgaben (deine Wahl) bis zum 15.2.2024 über Ilias ab.*

Aufgabe 1 – Deterministisches Schälen in Linearzeit

(a) Der Algorithmus constructByPeeling ist nichtdeterministisch in dem Sinne, dass nicht vorgeschrieben ist, welches i von der While-Schleife ausgewählt wird, wenn es mehrere zulässige Wahlen gibt. Zeige, dass der Erfolg des Algorithmus constructByPeeling nicht davon abhängt. Genauer noch: Für zwei mögliche Ausführungen von constructByPeeling bleibt die exakt gleiche Menge von Schlüsseln am Ende unplatziert.

(b) Verfeinere den Pseudocode so, dass ein deterministischer Algorithmus herauskommt, der erkennbar Laufzeit $O(n)$ hat. Nutze zur Unterstützung geeignete Datenstrukturen deiner Wahl.
Hinweis: Eine naheliegende Implementierung verwendet eine ungerichtete Graphdatenstruktur. Nimm an, dass diese folgendes leisten kann: (1) Einen (bipartiten) Graphen aus einer Kantenliste in Linearzeit aufbauen. (2) In Linearzeit über alle Knoten iterieren. (3) In $O(1)$ eine gegebene Kante löschen. (4) Den Grad eines Knotens in $O(1)$ bestimmen. (5) Für Grad 1 Knoten die inzidente Kante in $O(1)$ bestimmen.

Lösung 1

(a) Sei $X = (x_1, \ldots, x_k)$ die Folge von Schlüsseln, die eine vollständige Ausführung von constructByPeeling platziert (in dieser Reihenfolge) bevor die While-Schleife verlassen wird. Sei $Y = (y_1, \ldots, y_\ell)$ eine andere mögliche Folge. Es genügt zu zeigen, dass jedes Element, das in X vorkommt, auch in Y vorkommt: Aus Symmetriegründen folgt dann, dass X und Y die selben Elemente (möglicherweise in anderer Reihenfolge) enthalten.

Wir machen einen Beweis durch Widerspruch und nehmen an, $j \in [k]$ ist der kleinste Index sodass x_j nicht in Y vorkommt. Von der X-Ausführung wird x_j an einem Index i_j platziert, der für keinen der Schlüssel aus $S \setminus \{x_1, \ldots, x_{j-1}\}$ in Frage kommt. Also kommt der Index i_j auch für keinen Schlüssel aus $S \setminus \{y_1, \ldots, y_\ell\}$ in Frage, denn $\{y_1, \ldots, y_\ell\} \supseteq \{x_1, \ldots, x_{j-1}\}$ nach Wahl von j. Nach Annahme ist x_j am Ende der Y-Ausführung noch in S. Dann wäre aber i_j ein Index der nur für x_j und keinen anderen verbleibenden Schlüssel in Frage kommt. Damit wäre ein weiterer Durchlauf der While-Schleife möglich. Damit beschreibt Y keine vollständige Ausführung von constructByPeeling. Widerspruch.

(b) Wir legen dem Algorithmus den bipartiten Cuckoo-Graphen wie in der Vorlesung definiert zugrunde. Wir benutzen eine Warteschlange Q. Invariante ist, dass Q alle Tabellenknoten von Grad 1 enthält (und eventuell weitere Tabellenknoten). Es spielt keine Rolle ob Q eine FIFO/LIFO oder andere Art von Warteschlange ist.

Algorithm constructByPeeling-Linear(S, h_1, h_2, h_3)**:**
- $T \leftarrow [\bot, \dots, \bot]$ // leere Tabelle
- $G \leftarrow (S, [m], \{(x, h_i(x)) \mid x \in S, i \in [3]\})$ // cuckoo graph
- $Q \leftarrow \emptyset$ // leere Warteschlange
- **for** $i \in [m]$ **do**
  - **if** $\deg_G(i) = 1$ **then**
    - Q.push(i)
- **while not** Q.isEmpty **do**
  - $i \leftarrow Q$.pop
  - **if** $\deg_G(i) = 1$ **then**
    - $x \leftarrow$ eindeutiger Nachbar von i
    - $T[i] \leftarrow x$
    - $S \leftarrow S \setminus \{x\}$
    - **for** $j \in \{h_1(x), h_2(x), h_3(x)\}$ **do**
      - lösche die Kante (x, j) aus G
      - **if** $\deg_G(j) = 1$ **then**
        - Q.push(j)
- **if** $S = \emptyset$ **then**
  - **return** T
- **else**
  - **return** NOT-PEELABLE

Nach Annahme lässt sich G am Anfang in Zeit $O(m)$ konstruieren und jede weitere Einzeloperation lässt sich in $O(1)$ ausführen. Weil jeder Knoten nur einmal zu einem Grad 1 Knoten werden kann (weil wir nur Kanten löschen), wird für jedes $i \in [m]$ nur einmal Q.push(i) ausgeführt. Damit ist die Anzahl der While-Schleifendurchläufe auch höchstens m (weil da jeweils ein Element aus Q entfernt wird). Insgesamt ergibt sich $O(m)$.

Bemerkung. Anstatt eine Adjazenzlistendarstellung von G zu wählen, genügt im vorliegenden Fall auch eine sparsamere Darstellung. Das wäre aber eine eigene Übungsaufgabe.

Aufgabe 2 – Ausgedünnte Poissonverteilung

Sei $\lambda \in \mathbb{R}_+$ und $p \in [0, 1]$. Betrachte das zweistufige Zufallsexperiment in dem wir

1. zunächst $X \sim \text{Pois}(\lambda)$ sampeln und
2. dann $Y \sim \text{Bin}(X, p)$ sampeln.

Zeige: $Y \sim \text{Pois}(\lambda p)$.

Hinweis: Es genügt folgende Zutaten zusammenzustöpseln und etwas zu rechnen.

(i) Definition der Poissonverteilung $\Pr_{Z\sim\text{Pois}(\lambda)}[Z = i] = e^{-\lambda}\frac{\lambda^i}{i!}$.

(ii) Definition der Binomialverteilung $\Pr_{Z\sim\text{Bin}(n,p)}[Z = i] = \binom{n}{i}p^i(1-p)^{n-i}$.

(iii) Satz von der totalen Wahrscheinlichkeit: $\Pr[Y = i] = \sum_j \Pr[Y = i \mid X = j]\Pr[X = j]$.

(iv) Siehe Wikipedia: Characterizations_of_the_exponential_function

Lösung 2

Wir müssen zeigen, dass für alle $i \in \mathbb{N}_0$ gilt: $\Pr[Y = i] = e^{-\lambda p}\frac{(\lambda p)^i}{i!}$.

$$\begin{aligned}
\Pr[Y = i] &\overset{\text{(iii)}}{=} \sum_{j\geq 0} \Pr[Y = i \mid X = j]\Pr[X = j] = \sum_{j\geq 0} \Pr_{Y\sim\text{Bin}(j,p)}[Y = i]\Pr[X = j] \\
&\overset{\text{(i)}}{=} \sum_{j\geq i} \Pr_{Y\sim\text{Bin}(j,p)}[Y = i]e^{-\lambda}\frac{\lambda^j}{j!} \overset{\text{(ii)}}{=} \sum_{j\geq i}\binom{j}{i}p^i(1-p)^{j-i}e^{-\lambda}\frac{\lambda^j}{j!} \\
&= \sum_{j\geq i}\frac{p^i(1-p)^{j-i}e^{-\lambda}\lambda^j}{i!(j-i)!} = e^{-\lambda}\frac{(\lambda p)^i}{i!}\sum_{j\geq i}\frac{(1-p)^{j-i}\lambda^{j-i}}{(j-i)!} \\
&= e^{-\lambda}\frac{(\lambda p)^i}{i!}\sum_{j\geq 0}\frac{(1-p)^j\lambda^j}{j!} \overset{\text{(iv)}}{=} e^{-\lambda}\frac{(\lambda p)^i}{i!}e^{\lambda(1-p)} = e^{-\lambda p}\frac{(\lambda p)^i}{i!}.
\end{aligned}$$

Aufgabe 3 – Galton-Watson Prozesse und ein bisschen Biologie

Mitochondriale DNA wird von Müttern an ihre Kinder vererbt (der Vater spielt keine Rolle). In dieser Aufgabe betrachten wir die Wahrscheinlichkeit, dass eine bestimmte Mutation, die bei einer Frau (nennen wir sie Eva) auftritt, langfristig überlebt. Wir tun das in einem sehr einfachen Modell, das man Galton-Watson Prozess nennt: Wir nehmen an, dass jede Frau eine $\text{Pois}(2\lambda)$-verteilte Anzahl von Kindern hat, die die Mutation erben.

Sei G_λ der (möglicherweise) unendliche Baum, der entsteht, wenn man mit einem Wurzelknoten startet und jeder Knoten (unabhängig von allen anderen Knoten) eine $\text{Pois}(\lambda)$-verteilte Anzahl von Kindern erhält.

(a) Erkläre: Die Überlebenswahrscheinlichkeit der Mutation entspricht der Wahrscheinlichkeit, dass G_λ unendlich ist. Triff eine geeignete Annahme, um mit den Männern zu verfahren, und verwende Aufgabe 2.

(b) Sei p_i die Wahrscheinlichkeit, dass G_λ mindestens einen Knoten auf Ebene i hat. Drücke p_{i+1} in Abhängigkeit von p_i aus für $i \in \mathbb{N}_0$. Die Wurzelebene sei Ebene 0.
Hinweis: Betrachte zunächst ein Kind c der Wurzel. Was ist die Wahrscheinlichkeit, dass aus c ein Unterbaum erwächst, der bis Ebene $i + 1$ reicht? Nehme dann an, dass

die Wurzel X Kinder hat. Was ist (bedingt auf X) die Verteilung für die Anzahl $Y \in \{0, \dots, X\}$ derjenigen Kinder, aus denen ein Unterbaum erwächst, der bis Ebene $i+1$ reicht? Verwende dann Aufgabe 2.

(c) Bestimme für $\lambda \in \{0, 0.5, 1, 1.1, 1.5\}$ jeweils eine Approximationen für die Wahrscheinlichkeit s_λ, dass G_λ unendlich ist. Es bietet sich an Geogebra oder Wolfram Alpha zu nutzen und ähnlich vorzugehen wie in der Vorlesung (als wie eine Rekurrenz der Überlebenswahrscheinlichkeiten beim Schälen betrachtet haben).

Lösung 3

(a) Da Männer mitochondriale DNA nicht vererben können, sind Söhne für das langfristige Überleben oder Aussterben der Mutation irrelevant. Wir gehen davon aus, dass Kinder unabhängig voneinander mit Wahrscheinlichkeit $1/2$ männlich bzw. weiblich sind.[1] Wenn $Y \sim \text{Pois}(2\lambda)$ die Anzahl der Kinder einer Frau ist, dann gilt für die Anzahl Töchter also $X \sim \text{Bin}(Y, \frac{1}{2})$. Nach Aufgabe 2 gilt $X \sim \text{Pois}(\lambda)$.

Daher hat der Abstammungsbaum der Mutation startend bei Eva Verteilung G_λ. Es gilt

Die Mutation stirbt irgendwann aus $\Leftrightarrow$ G_λ hat endlich viele Ebenen $\Leftrightarrow$ G_λ ist endlich.

(b) Sei $\text{depth}(T) \in \mathbb{N}_0 \cup \{\infty\}$ die Tiefe eines Baumes T. Sei $X \sim \text{Pois}(\lambda)$ die Anzahl von Kindern der Wurzel von G_λ und $G_\lambda^{(1)}, G_\lambda^{(2)}, \dots, G_\lambda^{(X)}$ die bei diesen Kindern startenden Unterbäume. Sei $Y = |\{i \in \{1, \dots, X\} \mid \text{depth}(G_\lambda^{(i)}) \geq i-1\}|$ die Anzahl der Unterbäume mit Tiefe mindestens $i-1$. Weil die Unterbäume die selbe Verteilung wie G_λ haben, gilt $\Pr[\text{depth}(G_\lambda^{(i)}) \geq i-1] = p_{i-1}$ für alle $i \in \{1, \dots, X\}$. Weil Unterbäume unabhängig voneinander sind gilt ferner $Y \sim \text{Bin}(X, p_{i-1})$. Damit folgt $Y \sim \text{Pois}(\lambda p_{i-1})$ aus Aufgabe 2 und es gilt:

$$\begin{aligned} p_i &= \Pr[\text{depth}(G_\lambda) \geq i] = \Pr[\exists j \in \{1, \dots, X\} : \text{depth}(G_\lambda^{(j)}) \geq i-1] \\ &= \Pr[Y > 0] = 1 - \Pr[Y = 0] = 1 - e^{-\lambda p_{i-1}}. \end{aligned}$$

(c) Es gilt also $p_0 = 1$ und $p_{i+1} = 1 - e^{-\lambda p_i}$ für alle $i \geq 0$. Es bietet sich an, die Funktion $f(x) = 1 - e^{-\lambda x}$ zu betrachten. Iteriertes Anwenden der Funktion startend bei $x = 1$ generiert die Folge $(p_i)_{i \in \mathbb{N}}$. Wie in der Vorlesung ist f monoton fallend und unser Startwert erfüllt $f(p_0) < p_0$. Also ist die Folge monoton fallend. Sie muss gegen den größten Fixpunkt $p_\infty \in [0, 1]$ von f konvergieren. Die gesuchte Wahrscheinlichkeit ist $s_\lambda = p_\infty$. Für $\lambda \leq 1$ ist $f(x) = 1 - e^{-\lambda x} \leq 1 - e^{-x} \leq 1 - (1-x) = x$ mit Gleichheit genau für $x = 0$. Für solche λ gilt also $s_\lambda = p_\infty = 0$. Für $\lambda = 1.1$ und $\lambda = 1.5$ ist s_λ die positive Lösung von $x = 1 - e^{-1.1x}$ bzw. von $x = 1 - e^{-1.5x}$. Diese sind ungefähr $s_{1.1} = 0.176134$ und $s_{1.5} = 0.582812$.

[1] Das stimmt mindestens deshalb nicht wirklich, weil das Geschlecht von Zwillingen korreliert ist.

Aufgabe 4 – Verzerrte Binomialverteilung

Seien $\alpha, \beta \in \mathbb{R}_+$ sowie $c, d \in \mathbb{N}$ Konstanten und $X \sim \mathrm{Bin}(\lfloor \alpha m \rfloor - c, \beta/m)$. Zeige, dass sich X für $m \to \infty$ einer Poissonverteilung mit Parameter $\lambda = \alpha\beta$ annähert, das heißt zeige:

$$\forall i \in \mathbb{N} : \lim_{m\to\infty} \Pr_{X\sim\mathrm{Bin}(\lfloor \alpha m\rfloor - c, \beta/m)}[X = i] = \Pr_{Z\sim\mathrm{Pois}(\lambda)}[Z = i].$$

Hinweis: Siehe Hinweise (i), (ii), (iv) aus Aufgabe 2.

Lösung 4

Wir definieren zur Bequemlichkeit $c' = \lfloor \alpha m \rfloor - \alpha m + c$ sodass $\lfloor \alpha m \rfloor - c = \alpha m - c'$. Zwar ist c' nun von m abhängig, aber es gilt nach wie vor $|c'| = O(1)$.

$$\begin{aligned}
\Pr_{X\sim\mathrm{Bin}(\alpha m - c', \beta/m)}[X = i] &\overset{\text{(ii)}}{=} \binom{\alpha m - c'}{i} \left(\frac{\beta}{m}\right)^i \left(1 - \frac{\beta}{m}\right)^{\alpha m - c' - i} \\
&= \frac{(\alpha m - c') \cdot (\alpha m - c' - 1) \cdot \ldots \cdot (\alpha m - c' - i + 1)}{i!} \left(\frac{\beta}{m}\right)^i \left(1 - \frac{\beta}{m}\right)^{\alpha m - c' - i} \\
&= \underbrace{\left(\alpha - \frac{c'}{m}\right)}_{\to\alpha} \cdot \underbrace{\left(\alpha - \frac{c' - 1}{m}\right)}_{\to\alpha} \cdot \ldots \cdot \underbrace{\left(\alpha - \frac{c' - i + 1}{m}\right)}_{\to\alpha} \frac{\beta^i}{i!} \underbrace{\left(1 - \frac{\beta}{m}\right)^{\alpha m}}_{\overset{\text{(iv)}}{\to} e^{-\alpha\beta}} \underbrace{\left(1 - \frac{\beta}{m}\right)^{-c' - i}}_{\to 1} \\
&\overset{m\to\infty}{\longrightarrow} \alpha^i \frac{\beta^i}{i!} e^{-\alpha\beta} = \frac{\lambda^i}{i!} e^{-\lambda} \overset{\text{(i)}}{=} \Pr_{Z\sim\mathrm{Pois}(\lambda)}[X = i].
\end{aligned}$$

Der Vollständigkeit halber...

In der Vorlesung wurden noch folgende Aufgaben in Aussicht gestellt. Dafür reicht beim besten Willen leider die Zeit nicht. Für interessierte werden dennoch Lösungen bereitgestellt.

Aufgabe 5 – Peeling mit 2 Hashfunktionen

Angenommen wir verwenden den Schälalgorithmus constructByPeeling in einem Setting mit nur zwei Hashfunktionen h_1 und h_2. Wir gehen vereinfachend davon aus, dass $h_1(x) \neq h_2(x)$ für alle $x \in D$ gilt und unter dieser Einschränkung die Paare $(h_1(x), h_2(x))$ uniform zufällig und für verschiedene $x \in D$ unabhängig sind.[2] Zeige:

(a) Es werden alle Schlüssel platziert genau dann wenn der Graph

$$G = ([m], \{\{h_0(x), h_1(x)\} \mid x \in S\})$$

keine Kreise enthält.
Beachte: G ist als Multigraph zu verstehen, der mehrere Kopien derselben Kante enthalten darf (aber nicht muss).

(b) Falls $n = \Omega(m)$ ist, gibt es einen Kreis in G mit Wahrscheinlichkeit $\Omega(1)$.
Hinweis: Es genügt Kreise der Länge $k = 2$ zu betrachten (das heißt doppelte Kanten).

Lösung 5

(a) Wir können constructByPeeling als Algorithmus auf G auffassen, der wiederholt Knoten von Grad 1 sucht. Ist v ein solcher Knoten dessen inzidente Kante von einem Schlüssel x herkommt, dann wird x in v platziert und die Kante zu x wird gelöscht. Es werden also alle Schlüssel platziert, wenn das sukzessive Löschen von Kanten mit einem Endpunkt von Grad 1 mit dem leeren Graphen endet.

Starten wir mit einem Wald G, dann ist G in jedem Schritt ein Wald (löschen von Kanten aus einem Wald liefert wieder einen Wald). Solange der Wald nicht leer ist, also mindestens ein Baum mit mindestens einer Kante übrig ist, hat dieser ein Blatt von Grad 1 an dem weitergearbeitet werden kann. Ergo: Wenn G ein Wald ist, werden alle Schlüssel platziert.

Hat G dagegen mindestens einen Kreis, der von Schlüsseln $x_1, \ldots, x_k$ herkommt, dann kann keiner dieser Schlüssel der erste sein, der von constructByPeeling platziert wird, denn beide möglichen Positionen des Schlüsseln kommen auch für einen zyklisch benachbarten Schlüssel in Frage. Also werden nicht alle Schlüssel platziert.

(b) Wir fragen uns, wie wahrscheinlich eine Doppelkante in G ist. Wir können uns dazu vorstellen, dass wir beim Bestimmen der Hashwerte aller Schlüssel n mal mit Zurücklegen aus einer Urne mit $\binom{m}{2}$ Bällen ziehen, die den möglichen Kanten entsprechen.

[2] Um das sicherzustellen können wir zum Beispiel $h_2(x) := (h_1(x) + h_{\text{diff}}(x)) \bmod m$ definieren für ein $h_{\text{diff}} : D \to [m-1]$.

Dass wir paarweise verschiedene Bälle ziehen hat Wahrscheinlichkeit:

$$\begin{aligned}
&1 \cdot \left(1 - \frac{1}{\binom{m}{2}}\right) \cdot \left(1 - \frac{2}{\binom{m}{2}}\right) \cdot \ldots \cdot \left(1 - \frac{n-1}{\binom{m}{2}}\right) \\
&\leq 1 \cdot \left(1 - \frac{1}{m^2/2}\right) \cdot \left(1 - \frac{2}{m^2/2}\right) \cdot \ldots \cdot \left(1 - \frac{n-1}{m^2/2}\right) \\
&\leq \left(1 - \frac{n/2}{m^2/2}\right)^{n/2} = \left(1 - \frac{\alpha}{m}\right)^{\alpha m/2} \overset{m\to\infty}{\longrightarrow} e^{-\alpha^2/2}.
\end{aligned}$$

Im letzten Schritt nutzen wir $(1 + \frac{\lambda}{m})^m \to e^\lambda$ für $\lambda \in \mathbb{R}$ gilt. Die Wahrscheinlichkeit, dass wir nicht verschiedene Bälle ziehen und entsprechend eine Doppelkante in G vorkommt, ist damit im Grenzwert $1 - e^{-\alpha^2/2}$ und damit $\Omega(1)$. Insbesondere gibt es mit Wahrscheinlichkeit $\Omega(1)$ einen Kreis.

Aufgabe 6 – Der Schälalgorithmus bleibt nicht erst spät stecken[3]

Für eine Schlüsselmenge $S \subseteq D$ der Größe $n = |S|$ und Hashfunktionen $h_1, h_2, h_3 \sim \mathcal{U}([m]^D)$ betrachten wir den Cuckoo Graphen wie in der Vorlesung:

$$G = G_{S,h_1,h_2,h_3} = (S, [m], \{(x, h_i(x)) \mid x \in S, i \in [3]\})$$

Wir nehmen lediglich $\alpha = \frac{n}{m} \leq 1$ an. Sei $S' \subseteq S$ die Menge der Schlüssel, die vom Schälalgorithmus nicht entfernt werden können (es gilt $|S'| \in \{0, \ldots, n\}$). Wir wollen zeigen, dass $\Pr[|S'| \in \{1, \ldots, \delta m\}] = O(1/m)$ ist für eine Konstante $\delta > 0$ (die später gewählt wird).

Intution: Entweder gilt $S' = \varnothing$ oder $|S'|$ ist $\Omega(m)$; alles dazwischen ist unwahrscheinlich.

(a) Zeige: $|S'| = 1$ ist nicht möglich.

(b) Zeige: $|N(S')| \leq \frac{3}{2}|S'|$. Hierbei ist $N(S')$ die Menge aller Nachbarn von S' in G.
Hinweis: Betrachte den Subgraphen von G, der von S' und $N(S')$ induziert ist. Wieviele Kanten gibt es? Was ist der Grad der $N(S')$-Knoten?

(c) Zeige, dass es eine Konstante C gibt, sodass für $s \in \{2, \ldots, n\}$ folgendes gilt:

$$p_s := \Pr[\exists X \subseteq S, |X| = s : \exists Y \subseteq [m], |Y| = \lfloor \tfrac{3}{2} s \rfloor : N(X) \subseteq Y] \leq \left(C \cdot \frac{s}{m}\right)^{s/2}.$$

Hinweis: Einfach brutal Union-Bound über alle Möglichkeiten von X und Y verwenden. Nützlich ist außerdem die Abschätzung $\binom{n}{k} \leq (\frac{ne}{k})^k$ für Binomialkoeffizienten. Ignoriere die Gaussklammern, das machen alle so.

(d) Zeige (i) $p_2 + p_3 + p_4 + p_5 = O(1/m)$, (ii) $\sum_{s=6}^{\sqrt{m}} p_s = O(1/m)$, (iii) $\sum_{s=\sqrt{m}}^{m/(2C)} p_s = O(1/m)$.

[3]Diese Aufgabe ist etwas aufwändig.

(e) Wähle $\delta = 2C$ und zeige $\Pr[|S'| \in \{1, \dots, \delta n\}] \leq \sum_{s=2}^{\delta n} p_s = O(1/m)$.

Lösung 6

(a) Ein einzelner Schlüssel kann sich nicht selbst im Weg stehen.

(b) Je nachdem, ob man Doppelkanten zählt oder nicht, gehen von S' genau $3|S'|$ oder höchstens $3|S'|$ Kanten aus. Weil jeder Behälter in $N(S')$ von zwei verschiedenen Schlüsseln aus S' getroffen werden muss (sonst könnte man weiterschälen) hat jeder Knoten in $N(S')$ mindestens zwei Nachbarn in S'. Für die Anzahl e von Kanten zwischen S' und $N(S')$ gilt also $3|S'| \geq e \geq 2|N(S')|$. Umstellen liefert die Behauptung.

(c) Es gibt $\binom{n}{s}$ Möglichkeiten um X zu wählen und $\binom{m}{(3/2)s}$ Möglichkeiten um Y zu wählen. Für beliebiges $x \in X$ gilt $\Pr[N(\{x\}) \subseteq Y] = (|Y|/m)^3$, denn alle drei Hashwerte müssen (unabhängig voneinander) in der Menge Y landen. Wir erhalten somit folgendes (auf dem Weg fassen wir einige Konstanten durch ein $C = \Theta(1)$ zusammen):

$$\begin{aligned} p_s &= \Pr[\exists X \subseteq S, |X| = s : \exists Y \subseteq [m], |Y| = \lfloor \tfrac{3}{2}s \rfloor : N(X) \subseteq Y] \\ &\leq \binom{n}{s}\binom{m}{(3/2)s}\left(\frac{(3/2)\cdot s}{m}\right)^{3s} \leq \left(\frac{ne}{s}\right)^{s}\left(\frac{me}{(3/2)s}\right)^{(3/2)s}\left(\frac{(3/2)\cdot s}{m}\right)^{3s} \\ &\leq \left(\frac{n^2e^2}{s^2}\right)^{s/2}\left(\frac{2^3m^3e^3}{3^3s^3}\right)^{s/2}\left(\frac{3^6s^6}{2^6m^6}\right)^{s/2} \leq \left(\frac{Cm^2m^3s^6}{s^2s^3m^6}\right)^{s/2} = \left(\frac{Cs}{m}\right)^{s/2}. \end{aligned}$$

(d) (i) Es gilt $p_2 = O(m^{-1})$, $p_3 = O(m^{-3/2})$, $p_4 = O(m^{-2})$, $p_4 = O(m^{-5/2})$. Passt also.

(ii) Jeder der Summanden in $\sum_{s=6}^{\sqrt{m}} \left(\frac{Cs}{m}\right)^{s/2}$ ist höchstens $\left(\frac{C\sqrt{m}}{m}\right)^{6/2} = O(m^{-3/2})$. Da es $O(m^{1/2})$ Summanden gibt, summieren diese sie sich zu höchstens $O(m^{-1})$

(iii) Jeder der Summanden in $\sum_{s=\sqrt{m}}^{m/(2C)} \left(\frac{Cs}{m}\right)^{s/2}$ ist höchstens $\left(\frac{1}{2}\right)^{\sqrt{m}} = O(m^{-2})$ (*sehr* grob abgeschätzt). Da es $O(m)$ Summanden gibt, summieren diese sich zu höchstens $O(m^{-1})$.

(e) Wähle $\delta = \frac{1}{2C}$. Angenommen $|S'| = s$ für ein $s \in \{1, \dots, \delta m\}$. Sei $X = S'$ und $Y' := N(S')$. Nach (b) gilt $|Y'| \leq \frac{3}{2}s$ also gibt es eine Menge $Y \supseteq Y'$ mit $|Y| = \lfloor \frac{3}{2}s \rfloor$ und $N(S') \subseteq Y$. Die Mengen X und Y erfüllen das Ereignis dessen Wahrscheinlichkeit wir mit p_s beschränkt haben. Zusammenfassend können wir also schlussfolgern:

$$\Pr[|S'| \in \{1, \dots, \delta m\}] \leq \underbrace{\Pr[|S'| = 1]}_{0} + \sum_{s=2}^{\delta m} \Pr[|S'| = s] \leq \sum_{s=2}^{m/(2C)} p_s = O(1/m).$$