Karlsruher Institut für Technologie

Übung 1 – Algorithmen II

Yaroslav Akhremtsev, Demian Hespe – *yaroslav.akhremtsev@kit.edu, hespe@kit.edu*

Mit Folien von Michael Axtmann (teilweise)

`http://algo2.iti.kit.edu/AlgorithmenII_WS17.php`

Institut für Theoretische Informatik - Algorithmik II

Organisatorisches

Vorlesungen:

Mo 09:45–11:15 HS Neue Chemie
Di 15:45–16:30 HS Neue Chemie

Saalübung:

Di 16:30–17:15 HS Neue Chemie

Übungsblätter:

14-tägig, jeweils Dienstags, Musterlösung 9 Tage später
1. Blatt: 24.10.2017

Ilias Forum:

Fragen zum Vorlesungsinhalt (auch anonym möglich)

Vorlesungsaufzeichnung:

Mitschnitte der Vorlesung auf Youtube

Organisatorisches

Sprechstunden:

- Peter Sanders, Dienstag 13:45–14:45 Uhr, Raum 217
- Thomas Worsch, Freitag 8:00–9:00 Uhr, Raum 230
- Simon Gog, Nach Vereinbarung, Raum 220 (erst ab November)
- Demian Hespe, Nach Vereinbarung, Raum 210
- Yarsolav Akhremtsev, Donnerstag 14:00–15:00 Uhr, Raum 221

Letzte Vorlesung: 06. Februar 2018

Klausur: Mittwoch, 21. Februar 2018, 11:30 Uhr

Themenübersicht

Randomisierte Algorithmen

- Grundlagen
- Verifikation von Matrix-Matrix Multiplikation
- Coupon Collector Problem
- Harmonische Zahlen

Randomisierte Algorithmen

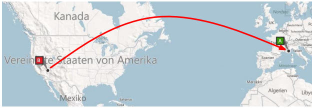
Übersicht

- *Las Vegas Algorithmus*
  - immer korrekte/optimale Lösung
  - Laufzeit ist Zufallsvariable
    $\rightarrow$ erwartete Laufzeit $\mathbb{E}[T]$
  - *Bsp.:* Quicksort

- *Monte Carlo Algorithmus*
  - falsche/suboptimale Lösung möglich
    $\rightarrow$ mit Wahrscheinlichkeit p
  - Beschränkte (worst-case) Laufzeit
  - *Bsp.:* Miller-Rabin Primzahltest,
    nicht vorbereiteter Student beim *multiple-choice* Test

Randomisierte Algorithmen

Las Vegas $\rightarrow$ Monte Carlo



geg: Las Vegas Algorithmus mit erwarteter Laufzeit $\mathbb{E}[T] = f(n)$
ges: Monte Carlo Algorithmus mit Laufzeit $\mathcal{O}(f(n))$, Fehlerrate p

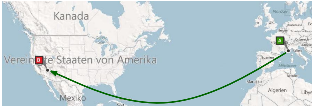
Idee: Abbruch nach Zeit $\alpha f(n)$

- Ausgabe FALSCH, wenn Algorithmus abgebrochen wurde
- $\mathbb{P}[T > \alpha f(n)] \leq 1/\alpha$ (Markov Ungleichung)

$\rightarrow$ Monte Carlo Algorithmus mit Laufzeit $\alpha f(n)$ und Fehlerrate $p = 1/\alpha$

Randomisierte Algorithmen

Monte Carlo → Las Vegas



geg: Monte Carlo Algorithmus mit Laufzeit $\mathcal{O}(f(n))$,
Fehlerrate $p < 1$ (gilt für alle Eingaben),
Korrektheit in $\mathcal{O}(g(n))$ prüfbar

ges: Las Vegas Algorithmus mit erwarteter Laufzeit $\mathbb{E}[T]$

Idee: Wiederhole MC bis korrekte Ergebnis gefunden

- Laufzeit $T \leq i \cdot \mathcal{O}(f(n) + g(n))$ (i Schritte benötigt)
- $\mathbb{E}[T] \leq \mathbb{E}[i] \cdot \mathcal{O}(f(n) + g(n))$
  - $\mathbb{E}[i] = \sum_{k=1}^{\infty} k \cdot p^{k-1} \cdot (1-p) = \frac{1}{1-p}$
    - Sei $f(p) = \sum_{k=0}^{\infty} \cdot p^k$ (also $f'(p) = \sum_{k=1}^{\infty} k \cdot p^{k-1}$)
    - $f(p) = \frac{1}{1-p}$ (geometrische Reihe)
    - $\Rightarrow f'(p) = \frac{1}{(1-p)^2}$
    - $\mathbb{E}[i] = f'(p) \cdot (1-p) = \frac{1}{1-p}$

→ Las Vegas Algorithmus mit erwarteter Laufzeit $\mathbb{E}[T] \leq \frac{\mathcal{O}(f(n)+g(n))}{1-p}$

Randomisierte Algorithmen

Matrix-Matrix Multiplikation

Aufgabe: Überprüfe, ob $X \cdot Y = Z$ für Matrizen X, Y, Z

- deterministisch:
  - berechne $X \cdot Y$ und vergleiche mit Z

  $\rightarrow$ Laufzeit $\mathcal{O}(n^3)$ (naiv), $\mathcal{O}(n^{2.37})$ (best)

- randomisiert:
  - Wähle 0-1 Vektor $r = (r_1, \ldots, r_n)$ zufällig
  - Wenn $X(Yr) = Zr$ dann KORREKT, sonst FALSCH

  $\rightarrow$ Laufzeit $\mathcal{O}(n^2)$

Behauptung: Wenn $XY \neq Z$ dann $\mathbb{P}[XYr = Zr] \leq 0.5$

Randomisierte Algorithmen

Matrix-Matrix Multiplikation

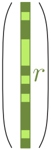
Aufgabe: Überprüfe, ob $X \cdot Y = Z$ für Matrizen X, Y, Z

- deterministisch:
  - berechne $X \cdot Y$ und vergleiche mit Z

  $\rightarrow$ Laufzeit $\mathcal{O}(n^3)$ (naiv), $\mathcal{O}(n^{2.37})$ (best)

- randomisiert:
  - Wähle 0-1 Vektor $r = (r_1, \ldots, r_n)$ zufällig
  - Wenn $X(Yr) = Zr$ dann KORREKT, sonst FALSCH

  $\rightarrow$ Laufzeit $\mathcal{O}(n^2)$



Behauptung: Wenn $XY \neq Z$ dann $\mathbb{P}[XYr = Zr] \leq 0.5$

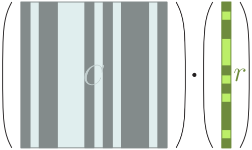
Randomisierte Algorithmen

Matrix-Matrix Multiplikation

Aufgabe: Überprüfe, ob $X \cdot Y = Z$ für Matrizen X, Y, Z

- deterministisch:
  - berechne $X \cdot Y$ und vergleiche mit Z
  - $\rightarrow$ Laufzeit $\mathcal{O}(n^3)$ (naiv), $\mathcal{O}(n^{2.37})$ (best)

- randomisiert:
  - Wähle 0-1 Vektor $r = (r_1, \ldots, r_n)$ zufällig
  - Wenn $X(Yr) = Zr$ dann KORREKT, sonst FALSCH
  - $\rightarrow$ Laufzeit $\mathcal{O}(n^2)$



Behauptung: Wenn $XY \neq Z$ dann $\mathbb{P}[XYr = Zr] \leq 0.5$

Randomisierte Algorithmen

Matrix-Matrix Multiplikation

Behauptung: Wenn $XY \neq Z$ dann $\mathbb{P}[XYr = Zr] \leq 0.5$

Definitionen:

- $A := XY$, $B := Z$

Annahme: $A \neq B$; Wann ist dann $Ar = Br$?

- $\exists i, j : A_{i,j} \neq B_{i,j}$
- Sei $a := A_i$ und $b := B_i$ (A_i: i'te Zeile von A)
  $\alpha := \sum_{k \neq j} a_k r_k$ und $\beta := \sum_{k \neq j} b_k r_k$
  $\Rightarrow ar = \alpha + a_j r_j$ und $br = \beta + b_j r_j$
  $\Rightarrow ar - br = (\alpha - \beta) + (a_j - b_j) r_j$
- Annahme: Werte für $r_{1 \ldots j-1, j+1 \ldots n}$ bereits festgelegt
  $\rightarrow$ noch 2 Möglichkeiten für r_j, Gleichheit bei maximal einer
  $\Rightarrow \mathbf{Pr}[ar - br = 0] = \mathbf{Pr}[r_j = -\frac{\alpha - \beta}{a_j - b_j}] \leq \frac{1}{2}$

$\rightarrow$ Fehlerrate $p \leq 0.5$

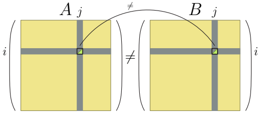
Randomisierte Algorithmen

Matrix-Matrix Multiplikation

Behauptung: Wenn $XY \neq Z$ dann $\mathbb{P}[XYr = Zr] \leq 0.5$

Definitionen:

- $A := XY$, $B := Z$



Annahme: $A \neq B$; Wann ist dann $Ar = Br$?

- $\exists i, j : A_{i,j} \neq B_{i,j}$
- Sei $a := A_i$ und $b := B_i$ (A_i: i'te Zeile von A)
  $\alpha := \sum_{k \neq j} a_k r_k$ und $\beta := \sum_{k \neq j} b_k r_k$
  $\Rightarrow ar = \alpha + a_j r_j$ und $br = \beta + b_j r_j$
  $\Rightarrow ar - br = (\alpha - \beta) + (a_j - b_j) r_j$
- Annahme: Werte für $r_{1 \ldots j-1, j+1 \ldots n}$ bereits festgelegt
  $\rightarrow$ noch 2 Möglichkeiten für r_j, Gleichheit bei maximal einer
  $\Rightarrow \mathbf{Pr}[ar - br = 0] = \mathbf{Pr}[r_j = -\frac{\alpha - \beta}{a_j - b_j}] \leq \frac{1}{2}$

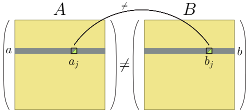
$\rightarrow$ Fehlerrate $p \leq 0.5$

Behauptung: Wenn $XY \neq Z$ dann $\mathbb{P}[XYr = Zr] \leq 0.5$

Definitionen:

- $A := XY$, $B := Z$

Annahme: $A \neq B$; Wann ist dann $Ar = Br$?



- $\exists i, j : A_{i,j} \neq B_{i,j}$
- Sei $a := A_i$ und $b := B_i$ (A_i: i'te Zeile von A)
  $\alpha := \sum_{k \neq j} a_k r_k$ und $\beta := \sum_{k \neq j} b_k r_k$
  $\Rightarrow ar = \alpha + a_j r_j$ und $br = \beta + b_j r_j$
  $\Rightarrow ar - br = (\alpha - \beta) + (a_j - b_j) r_j$
- Annahme: Werte für $r_{1\ldots j-1, j+1 \ldots n}$ bereits festgelegt
  $\rightarrow$ noch 2 Möglichkeiten für r_j, Gleichheit bei maximal einer
  $\Rightarrow \mathbf{Pr}[ar - br = 0] = \mathbf{Pr}[r_j = -\frac{\alpha - \beta}{a_j - b_j}] \leq \frac{1}{2}$

$\rightarrow$ Fehlerrate $p \leq 0.5$

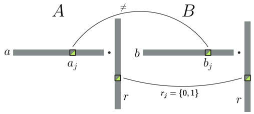
Randomisierte Algorithmen

Matrix-Matrix Multiplikation

Behauptung: Wenn $XY \neq Z$ dann $\mathbb{P}[XYr = Zr] \leq 0.5$

Definitionen:

- $A := XY$, $B := Z$



Annahme: $A \neq B$; Wann ist dann $Ar = Br$?

- $\exists i, j : A_{i,j} \neq B_{i,j}$
- Sei $a := A_i$ und $b := B_i$ (A_i: i'te Zeile von A)
  $\alpha := \sum_{k \neq j} a_k r_k$ und $\beta := \sum_{k \neq j} b_k r_k$
  $\Rightarrow ar = \alpha + a_j r_j$ und $br = \beta + b_j r_j$
  $\Rightarrow ar - br = (\alpha - \beta) + (a_j - b_j) r_j$
- Annahme: Werte für $r_{1 \ldots j-1, j+1 \ldots n}$ bereits festgelegt
  $\rightarrow$ noch 2 Möglichkeiten für r_j, Gleichheit bei maximal einer
  $\Rightarrow \mathbf{Pr}[ar - br = 0] = \mathbf{Pr}[r_j = -\frac{\alpha - \beta}{a_j - b_j}] \leq \frac{1}{2}$

$\rightarrow$ Fehlerrate $p \leq 0.5$

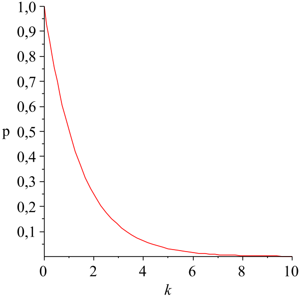
Randomisierte Algorithmen

Matrix-Matrix Multiplikation

Beschleunigung durch *probability boosting*

(nur bei $p \leq 0.5$ schnelle Konvergenz)

- Wiederhole Test k mal mit unterschiedlicher Wahl von r
  - Ein Test liefert FALSCH
    $\rightarrow AB \neq C$, fertig
  - Alle Tests liefern KORREKT
    $\rightarrow$ *false positive* mit Wahrscheinlichkeit
    $\mathbb{P}[ABr = Cr] \leq 0.5^k$



$\rightarrow$ Laufzeit $\mathcal{O}(kn^2)$

(linear längere Laufzeit bei exponentiell weniger Fehler)

Coupon Collector

Müslipackungen enthalten jeweils eine von n verschiedenen Sammelkarten. Wie viele Packungen muss ich kaufen um alle Karten beisammenzuhaben?

- X = # Packungen bis mind. eine von jeder Karte
- $X = \sum_{i=1}^{n} X_i$, mit X_i = # Packungen während ich $i-1$ Karten hatte
  - X_i sind geometrische Zufallsvariablen mit $p_i = 1 - \frac{i-1}{n}$
  - $\mathbb{E}[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}$
- $\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbb{E}[X_i]$ (Linearität des Erwartungswertes)
- $= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i}$

Coupon Collector

Müslipackungen enthalten jeweils eine von n verschiedenen Sammelkarten. Wie viele Packungen muss ich kaufen um alle Karten beisammenzuhaben?

- X = # Packungen bis mind. eine von jeder Karte
- $X = \sum_{i=1}^{n} X_i$, mit X_i = # Packungen während ich $i-1$ Karten hatte
  - X_i sind geometrische Zufallsvariablen mit $p_i = 1 - \frac{i-1}{n}$
  - $\mathbb{E}[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}$
- $\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbb{E}[X_i]$ (Linearität des Erwartungswertes)
- $= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i}$

Coupon Collector

Müslipackungen enthalten jeweils eine von n verschiedenen Sammelkarten. Wie viele Packungen muss ich kaufen um alle Karten beisammenzuhaben?

- X = # Packungen bis mind. eine von jeder Karte
- $X = \sum_{i=1}^{n} X_i$, mit X_i = # Packungen während ich $i-1$ Karten hatte
  - X_i sind geometrische Zufallsvariablen mit $p_i = 1 - \frac{i-1}{n}$
  - $\mathbb{E}[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}$
- $\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbb{E}[X_i]$ (Linearität des Erwartungswertes)
- $= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i}$

Coupon Collector

Müslipackungen enthalten jeweils eine von n verschiedenen Sammelkarten. Wie viele Packungen muss ich kaufen um alle Karten beisammenzuhaben?

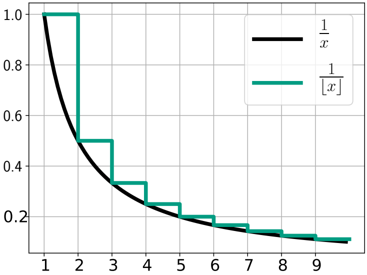
- X = # Packungen bis mind. eine von jeder Karte
- $X = \sum_{i=1}^{n} X_i$, mit X_i = # Packungen während ich $i-1$ Karten hatte
  - X_i sind geometrische Zufallsvariablen mit $p_i = 1 - \frac{i-1}{n}$
  - $\mathbb{E}[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}$
- $\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbb{E}[X_i]$ (Linearität des Erwartungswertes)
- $= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i}$

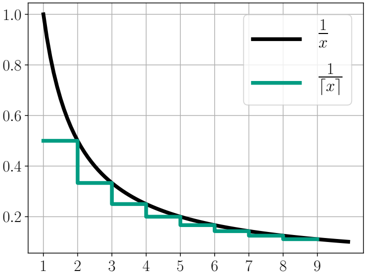
Harmonische Zahlen

- $H_n = \sum_{i=1}^{n} \frac{1}{i}$

$$\ln n = \int_{x=1}^{n} \frac{1}{x} dx \leq \sum_{i=1}^{n} \frac{1}{i}$$

$$\sum_{i=2}^{n} \frac{1}{i} \leq \int_{x=1}^{n} \frac{1}{x} dx = \ln n$$





- $\ln n \leq H_n \leq \ln n + 1$

Coupon Collector

Müslipackungen enthalten jeweils eine von n verschiedenen Sammelkarten. Wie viele Packungen muss ich kaufen um alle Karten beisammenzuhaben?

- X = # Packungen bis mind. eine von jeder Karte
- $X = \sum_{i=1}^{n} X_i$, mit X_i = # Packungen während ich $i-1$ Karten hatte
  - X_i sind geometrische Zufallsvariablen mit $p_i = 1 - \frac{i-1}{n}$
  - $\mathbb{E}[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}$
- $\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbb{E}[X_i]$ (Linearität des Erwartungswertes)
- $= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i} = nH_n \leq n \ln n + n$

Coupon Collector - Tailbound

- X = # Packungen bis mind. eine von jeder Karte
- $X = \sum_{i=1}^{n} X_i$, mit X_i = # Packungen während ich $i-1$ Karten hatte
  - X_i sind geometrische Zufallsvariablen mit $p_i = 1 - \frac{i-1}{n}$
  - $\mathbb{E}[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}$
  - $\mathbf{Var}[X_i] = \frac{1-p_i}{p_i^2} \leq \frac{1}{p_i^2}$
- $\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbb{E}[X_i]$ (Linearität des Erwartungswertes)
- $= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i} = nH_n \leq n \ln n + n$
- Chebyshev: $\mathbb{P}(|X - \mathbb{E}[X]| \geq t \cdot \mathbb{E}[X]) \leq \frac{\mathbf{Var}[X]}{t^2 \cdot (\mathbb{E}[X])^2}$
- $\mathbf{Var}[X] = \sum_{i=1}^{n} \mathbf{Var}[X_i] \leq \sum_{i=1}^{n} (\frac{n}{n-i+1})^2 = n^2 \sum_{i=1}^{n} (\frac{1}{i^2}) \leq \frac{\pi^2 n^2}{6}$
- $\mathbb{P}(|X - nH_n| \geq nH_n) \leq \frac{n^2 \pi^2 / 6}{(nH_n)^2} = \frac{\pi^2}{6H_n^2} \in \mathcal{O}(\frac{1}{\ln^2 n})$