Mathematische Analyse randomisierter Sortieralgorithmen

Masterarbeit
am Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg

vorgelegt von
Johanna Wiehe

September 2017
Betreuer: Prof. Dr. Volkmar Welker

Inhaltsverzeichnis

Einleitung

In dieser Arbeit werden diverse Sortieralgorithmen mit elementaren, insbesondere probabilistischen Methoden der diskreten Mathematik analysiert. Alle hier betrachteten Verfahren haben das Ziel, eine gegebene Liste endlich vieler Elemente einer Menge, auf der eine totale oder streng totale Ordnung definiert ist, gemäß dieser Ordnung zu sortieren. Ohne Einschränkung kann man also von einer endlichen Liste nicht notwendigerweise verschiedener natürlicher Zahlen ausgehen, die der Größe nach aufsteigend sortiert werden sollen.
Nun ist es bei manchen Verfahren vorteilhaft, zufällige Komponenten einzubauen, um etwa wiederholtes schlechtes Wählen von Entscheidungsvariablen zu vermeiden. Der Schwerpunkt wird in dieser Arbeit allerdings nicht auf diesen sogenannten randomisierten Algorithmen selbst sondern auf deren Analyse liegen, die naheliegenderweise auch einem probabilistischen Ansatz folgt.
Da jeder hier betrachtete Algorithmus vergleichsbasiert und fast jeder mithilfe von Vertauschungen arbeitet, wird als Maß der Effizienz die Anzahl der benötigten Vergleiche und gegebenenfalls auch die der getätigten Vertauschungen herangezogen.
Die Methoden, diese Größen zu bestimmen, sollten mit dem mathematischen Grundlagenwissen aus der Analysis und der Linearen Algebra bekannt oder zumindest nachvollziehbar sein. Alle weiteren zum Verständnis benötigten Grundlagen und Notationen werden im ersten Kapitel, teilweise auch mit Beweisen, bereitgestellt.
Im zweiten Kapitel wird der stark randomisierte Sortieralgorithmus *Bogosort* vorgestellt. Beim Überprüfen auf Sortiertheit werden drei unterschiedliche Szenarien betrachtet und miteinander verglichen. Zudem wird sich hier die Frage der Terminierung des Algorithmus stellen. Insgesamt orientiert sich die Analyse in diesem Kapitel an der unter [3] angegebenen Quelle, wobei hier vorwiegend Erklärungen ergänzt, Beweise vervollständigt und gegebenenfalls korrigiert werden.
Diesem sehr langsamen, d.h. ineffizienten Verfahren wird dann im dritten Kapitel der sehr viel effizientere Algorithmus *Quicksort* entgegengestellt und auf zweierlei Arten analysiert. Dabei wird der probabilistische Anteil einmal im Algorithmus selbst und einmal ausschließlich in der Analyse zu finden sein. Als Quelle ist hier vorwiegend [7] zu nennen.
In den nachfolgenden Kapiteln werden dann sowohl das nicht randomisierte Verfahren *Bubblesort* als auch der im zweiten Kapitel zu behandelnde, stark vom Zufall abhängige Algorithmus *Bogosort* unter zweier probabilistischer Annahmen untersucht. Die üblicherweise getroffene Annahme einer gleich-

verteilt zufällig gegebenen Liste im Input wird durch die Voraussetzung einer mittels der im ersten Kapitel einzuführenden *Plackett-Luce-Verteilung* gegebenen Eingabe verallgemeinert, sodass die Aussagen der jeweils vorigen Kapitel einerseits präzisiert werden und andererseits eine allgemeinere Anwendung finden können. Die vorliegende Arbeit liefert in diesem Aspekt erste Einblicke, Vermutungen und Herangehensweisen, die durchaus fortgeführt werden können.

1 Notation und Grundlagen

In diesem Abschnitt werden zunächst die für diese Arbeit wichtigen Grundlagen aus der diskreten Mathematik und der elementaren Stochastik zusammengestellt sowie Notationen festgelegt.

1.1 Grundlagen aus der diskreten Mathematik

Beim Sortieren einer endlichen Liste paarweise verschiedener Zahlen, die einer streng totalen Ordnung unterliegen, bietet es sich an diese mit einer bijektiven Abbildung

$$\sigma : \{1, ..., n\} \to \{1, ..., n\}$$

zu identifizieren. Eine solche Abbildung heißt **Permutation**, die in dieser Arbeit häufig als sogenanntes **Wort**, geschrieben in der Notation

$$\sigma(1)\sigma(2)\cdots\sigma(n),$$

Verwendung finden wird. Die Menge aller Permutationen auf $\{1, ..., n\}$ wird mit S_n bezeichnet und bildet zusammen mit der Hintereinanderausführung eine Gruppe, die **symmetrische Gruppe**.

Naheliegenderweise ist das Ziel beim Sortieren einer gegebenen Permutation, diese in die **Identität**, hier mit id bezeichnet und als Wort mit $1 \cdots n$ notiert, zu überführen. Dabei wird die Anzahl der Fehlstände, auch Inversionen genannt, eine entscheidende Rolle spielen. Wir nennen ein Paar (i, j) mit $1 \leq i < j \leq n$ **Inversion** der Permutation $\sigma \in S_n$, falls $\sigma(i) > \sigma(j)$ ist und schreiben

$$Inv(\sigma) := \big\{(i, j) | (i, j) \text{ ist Inversion von } \sigma\big\}$$

für die Menge der Inversionen von σ.

Beispiel 1.1

(i) *Offenbar hat die Identität keinerlei Inversionen, daher gilt*

$$\#Inv(id) = 0.$$

(ii) *Die Permutation mit den meisten Inversionen ist* $n(n-1)\cdots 1$. *Es gilt*

$$\#Inv\big(n(n-1)\cdots 1\big) = \sum_{k=1}^{n-1} k = \frac{1}{2}(n-1)n.$$

(iii) *Die Anzahl der Permutationen mit genau einer Inversion ist* $n-1$. *Betrachtet man nämlich eine Permutation* $\sigma \in S_n$ *mit genau einer Inversion* (i,j), $1 \leq i < j \leq n$, *so muss* $j = i+1$ *gelten, andernfalls gäbe es mehr als einen Fehlstand. Da* $i \in \{1, ..., n-1\}$ *gilt, gibt es genau* $n-1$ *Möglichkeiten* i *und somit auch* j *zu platzieren und wir erhalten*

$$\#\{\sigma \in S_n | \#Inv(\sigma) = 1\} = n-1.$$

1.2 Grundlagen aus der elementaren Stochastik

Da die meisten in dieser Arbeit untersuchten Algorithmen einen randomisierten Teil beinhalten, bietet es sich an, auch deren Analyse probabilistisch auszurichten. Dazu definieren wir zunächst einen diskreten Wahrscheinlichkeitsraum:

Ist Ω eine nichtleere, endliche oder abzählbar unendliche Menge und $\mathbb{P} : \Omega \to [0,1]$ eine Abbildung mit

$$\sum_{\omega \in \Omega} \mathbb{P}(\omega) = 1,$$

so heißt $(\Omega, \mathbb{P})$ ein **diskreter Wahrscheinlichkeitsraum**. Die Elemente von Ω heißen **Ergebnisse**, die Teilmengen von Ω **Ereignisse**, die Abbildung $\mathbb{P}$ heißt **Wahrscheinlichkeitsverteilung**. Die Wahrscheinlichkeit eines Ereignisses $A \subset \Omega$ wird durch

$$\mathbb{P}(A) := \sum_{\omega \in A} \mathbb{P}(\omega)$$

definiert. Alle in dieser Arbeit betrachteten Wahrscheinlichkeitsräume werden ausschließlich diskret sein.

Beispiel 1.2

(i) *Für eine endliche Menge Ω mit $\#\Omega = n$ wird durch $\mathbb{P} : \Omega \to [0,1]$ mit $\mathbb{P}(\omega) := \frac{1}{n}$, für $\omega \in \Omega$, eine Wahrscheinlichkeitsverteilung, die **Gleichverteilung**, definiert, da*

$$\sum_{\omega\in\Omega} \mathbb{P}(\omega) = \sum_{\omega\in\Omega} \frac{1}{n} = 1$$

gilt. Somit bildet $(\Omega, \mathbb{P})$ einen diskreten Wahrscheinlichkeitsraum.

(ii) *Im sogenannten **Bernoulli-Experiment** gibt es nur zwei mögliche Ausgänge: Erfolg und Misserfolg. Dabei gibt $p \in [0,1]$ die Wahrscheinlichkeit für einen Erfolg (hier durch 1 modelliert) an, und $\Omega := \{0,1\}$ und $\mathbb{P} : \Omega \to [0,1]$ mit $\mathbb{P}(0) := 1 - p$ und $\mathbb{P}(1) := p$ bilden offenbar einen diskreten Wahrscheinlichkeitsraum.*

(iii) *Für reelle Zahlen $x_1, ..., x_n > 0$ bildet die symmetrische Gruppe S_n zusammen mit $\mathbb{P}_x : S_n \to [0,1]$,*

$$\mathbb{P}_x(\sigma) := \prod_{i=1}^{n} \frac{x_{\sigma(i)}}{x_{\sigma(i)} + ... + x_{\sigma(n)}}$$

einen diskreten Wahrscheinlichkeitsraum. Dass dabei $\sum_{\sigma\in S_n} \mathbb{P}_x(\sigma) = 1$ gilt kann per Induktion über n gezeigt werden:
Der Induktionsanfang ist klar, für den Induktionsschritt definieren wir $S_{n,k} := \{\sigma \in S_n | \sigma(1) = k\}$ und erhalten für $x_1, ..., x_{n+1} > 0$

$$\begin{aligned}
\sum_{\sigma\in S_{n+1}} \mathbb{P}_x(\sigma) &= \sum_{\sigma\in S_{n+1}} \frac{x_{\sigma(1)} \cdots x_{\sigma(n+1)}}{\prod_{i=1}^{n+1}(x_{\sigma(i)} + ... + x_{\sigma(n+1)})} \\
&= \sum_{k=1}^{n+1} \sum_{\sigma\in S_{n+1,k}} \frac{x_1 \cdots x_{n+1}}{\prod_{i=1}^{n+1}(x_{\sigma(i)} + ... + x_{\sigma(n+1)})} \\
&= \sum_{k=1}^{n+1} \left(\frac{x_k}{x_1 + ... + x_{n+1}} \sum_{\sigma\in S_{n+1,k}} \frac{x_1 \cdots x_{k-1}x_{k+1} \cdots x_{n+1}}{\prod_{i=2}^{n+1}(x_{\sigma(i)} + ... + x_{\sigma(n+1)})} \right) \\
&\overset{IV}{=} \sum_{k=1}^{n+1} \frac{x_k}{x_1 + ... + x_{n+1}} = 1.
\end{aligned}$$

*Folglich ist $\mathbb{P}_x$ tatsächlich eine Wahrscheinlichkeitsverteilung, die sogenannte **Plackett-Luce-Verteilung**, die speziell mit $x_i = x^i, 1 \leq i \leq n$, ab Kapitel 4.2 aufgegriffen werden wird.*

Zur mathematischen Modellierung der Ergebnisse eines Zufallsexperiments definieren wir: Eine reelle **Zufallsvariable** X ist eine Abbildung $X : \Omega \to \mathbb{R}$. Die einzelnen Werte von $X(\Omega)$ heißen **Realisierungen** und mit

$$\mathbb{P}_X(z) := \mathbb{P}\big(\{\omega \in \Omega | X(\omega) = z\}\big)$$

induziert X auf natürliche Weise einen diskreten Wahrscheinlichkeitsraum auf $X(\Omega)$. Für $\mathbb{P}_X(z)$ schreiben wir auch $\mathbb{P}\big(\{X = z\}\big)$ und analog schreiben wir $\mathbb{P}\big(\{X \geq z\}\big)$ abkürzend für $\mathbb{P}\big(\{\omega \in \Omega | X(\omega) \geq z\}\big)$.

Beispiel 1.3 *Sei* $(\Omega, \mathbb{P})$ *ein diskreter Wahrscheinlichkeitsraum.*

(i) Ist $A \subset \Omega$ *ein Ereignis, so gibt die* ***Indikatorzufallsvariable*** $\mathbb{1}_A : \Omega \to \{0, 1\}$ *an, ob dieses Ereignis eintritt. Sie wird wie folgt definiert*

$$\mathbb{1}_A(\omega) := \begin{cases} 1 & \text{, falls } \omega \in A, \\ 0 & \text{, falls } \omega \notin A. \end{cases}$$

(ii) Ist das Bild einer Zufallsvariable $X : \Omega \to \mathbb{R}$ *endlich, d.h. es gibt* $x_1, ..., x_n \in \mathbb{R}$, *sodass* $X(\Omega) = \{x_1, ..., x_n\}$, *und tritt jede dieser Realisierungen mit gleicher Wahrscheinlichkeit ein, also*

$$\mathbb{P}\big(\{X = x\}\big) = \frac{1}{n} \text{ für alle } x \in X(\Omega),$$

so nennen wir die Zufallsvariable X ***gleichverteilt****.*

(iii) Für eine unendliche Folge unabhängiger Bernoulli-Experimente definieren wir die Zufallsvariable $X : \Omega \to \mathbb{N}_0$, *die die Anzahl der Misserfolge angibt, die vor dem ersten Erfolg auftreten. Gibt* $p \in (0, 1)$ *die Wahrscheinlichkeit für einen Erfolg an, so gilt*

$$\mathbb{P}\big(\{X = x\}\big) = p(1-p)^x \text{ für alle } x \in X(\Omega)$$

und wir nennen X ***geometrisch*** *verteilt auf* $\mathbb{N}_0$.

Kommen wir nun zu dem wichtigen Konzept der stochastischen Unabhängigkeit, hier zunächst von Ereignissen.
Ist I eine nichtleere Indexmenge und sind $(A_i)_{i \in I}$ Ereignisse im Wahrscheinlichkeitsraum $(\Omega, \mathbb{P})$, so heißen $(A_i)_{i \in I}$ **unabhängig**, falls für alle endlichen, nichtleeren Indexmengen $J \subset I$ gilt

$$\mathbb{P}\Big(\bigcap_{j \in J} A_j\Big) = \prod_{j \in J} \mathbb{P}(A_j).$$

Wir nennen zwei Zufallsvariablen $X, Y : \Omega \to \mathbb{R}$ **unabhängig**, falls

$$\mathbb{P}\big(\{X = x\} \cap \{Y = y\}\big) = \mathbb{P}\big(\{X = x\}\big) \cdot \mathbb{P}\big(\{Y = y\}\big)$$

für alle $x \in X(\Omega)$ und $y \in Y(\Omega)$ gilt.

Eine wichtige Kenngröße von Zufallsvariablen ist die Zahl, die sie umgangssprachlich im Mittel annehmen. Formal ist der **Erwartungswert** $\mathbb{E}[X]$ einer Zufallsvariable X auf einem Wahrscheinlichkeitsraum $(\Omega, \mathbb{P})$ definiert, falls $\sum_{\omega \in \Omega} |X(\omega)| \mathbb{P}(\omega)$ in $\mathbb{R}$ existiert. In diesem Fall setzt man

$$\mathbb{E}[X] := \sum_{\omega \in \Omega} X(\omega) \mathbb{P}(\omega).$$

Lemma 1.4 *Ist $X : \Omega \to \mathbb{R}$ eine Zufallsvariable, dann gilt*

(i) $\mathbb{E}[X] = \sum_{x \in X(\Omega)} x \cdot \mathbb{P}_X(x)$ *und*

(ii) ist zusätzlich $X(\Omega) \subset \mathbb{N}_0$, so gilt $\mathbb{E}[X] = \sum_{x \geq 1} \mathbb{P}\big(\{X \geq x\}\big)$.

Beweis :

(i) $$\mathbb{E}[X] = \sum_{\omega \in \Omega} X(\omega) \mathbb{P}(\omega) = \sum_{x \in X(\Omega)} \Big(\sum_{\omega = X^{-1}(x)} \mathbb{P}(\omega) \Big) x = \sum_{x \in X(\Omega)} \mathbb{P}\big(\{X = x\}\big) x$$
$$= \sum_{x \in X(\Omega)} \mathbb{P}_X(x) \cdot x$$

(ii) $$\mathbb{E}[X] = \sum_{x \geq 0} \mathbb{P}\big(\{X = x\}\big) x = \sum_{x \geq 1} \mathbb{P}\big(\{X = x\}\big) x = \sum_{x \geq 1} \sum_{z=1}^{x} \mathbb{P}\big(\{X = x\}\big)$$
$$= \sum_{z \geq 1} \sum_{x \geq z} \mathbb{P}\big(\{X = x\}\big) = \sum_{z \geq 1} \mathbb{P}\big(\{X \geq z\}\big)$$

$\square$

Beispiel 1.5

(i) Ist $\mathbb{1}_A$ die Indikatorzufallsvariable eines Ereignisses $A \subset \Omega$ im Wahrscheinlichkeitsraum $(\Omega, \mathbb{P})$, so gilt für ihren Erwartungswert

$$\mathbb{E}(\mathbb{1}_A) = \sum_{\omega \in \Omega} \mathbb{1}_A(\omega) \mathbb{P}(\omega) = \mathbb{P}\big(\{\mathbb{1}_A = 1\}\big) = \mathbb{P}(A).$$

(ii) Der Erwartungswert einer geometrisch verteilten Zufallsvariable X mit Erfolgswahrscheinlichkeit $p \in (0,1)$ berechnet sich mithilfe der geometrischen Reihe wie folgt:

$$\begin{aligned}\mathbb{E}[X] &= \sum_{k\geq 1} kp(1-p)^{k-1} = \sum_{k\geq 0}(k+1)p(1-p)^k \\ &= \sum_{k\geq 0} kp(1-p)^k + \sum_{k\geq 0} p(1-p)^k \\ &= (1-p)\mathbb{E}[X] + \frac{p}{1-(1-p)}.\end{aligned}$$

Durch Äquivalenzumformung erhält man somit $\mathbb{E}[X] = \frac{1}{p}$.

Satz 1.6 *Sei $(\Omega, \mathbb{P})$ ein Wahrscheinlichkeitsraum und $X, Y : \Omega \to \mathbb{R}$ Zufallsvariablen. Dann gilt:*

(i) Der Erwartungswert ist linear, d.h. für $a, b \in \mathbb{R}$ gilt

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y].$$

(ii) Sind X und Y zudem unabhängig, so gilt für deren Produkt im Erwartungswert

$$\mathbb{E}[X \cdot Y] = \mathbb{E}[X] \cdot \mathbb{E}[Y].$$

Die Beweise erfolgen durch einfaches Einsetzen und können in [5] nachgeschlagen werden. Dasselbe gilt für den Beweis der folgenden, grundlegenden Ungleichung.

Satz 1.7 ***(Markov-Ungleichung)***
Sei $(\Omega, \mathbb{P})$ ein Wahrscheinlichkeitsraum und $X : \Omega \to \mathbb{R}$ eine Zufallsvariable, die nur nichtnegative Werte annimmt. Dann gilt für alle $t \in \mathbb{R}$ mit $t > 0$

$$\mathbb{P}\big(\{X \geq t\}\big) \leq \frac{\mathbb{E}[X]}{t}.$$

Zur besseren Berechnung von Erwartungswerten wird das Konzept von *Stoppzeiten* nützlich sein:

Sei $(\Omega, \mathbb{P})$ ein Wahrscheinlichkeitsraum und $(X_i)_{i\geq 1}$ eine Folge von Zufallsvariablen mit $\mathbb{E}[X_i] < \infty$ für alle $i \geq 1$. Die Zufallsvariable $N : \Omega \to \mathbb{N}_0$

heißt **Stoppzeit** für die Folge $(X_i)_{i\geq 1}$, falls $\mathbb{E}[N] < \infty$ und die Zufallsvariable $\mathbb{1}_{\{N\leq n\}}$ unabhängig von $(X_i)_{i>n}$ ist, für alle $n \in \mathbb{N}_0$.

Damit können wir nun diesen später sehr hilfreichen Satz, auch als die *Formel von Wald* bekannt, formulieren.

Satz 1.8 ***(Wald)*** *Sei $(X_i)_{i\geq 1}$ eine Folge unabhängiger, identisch verteilter Zufallsvariablen mit $\mathbb{E}[X_1] < \infty$ und N eine Stoppzeit für diese Folge. Dann gilt für $S(N) = \sum_{i=1}^{N} X_i$*

$$\mathbb{E}[S(N)] = \mathbb{E}[X_1] \cdot \mathbb{E}[N].$$

Beweis : Zunächst gilt $S(N) = \sum_{i=1}^{N} X_i = \sum_{i=1}^{\infty} X_i \mathbb{1}_{\{N\geq i\}}$ und mit der Linearität des Erwartungswertes

$$\mathbb{E}[S(N)] = \sum_{i=1}^{\infty} \mathbb{E}[X_i \cdot \mathbb{1}_{\{N\geq i\}}].$$

Nun ist X_i unabhängig von $\mathbb{1}_{\{N\geq i\}}$, da N eine Stoppzeit ist und somit $(X_i)_{i>n}$ unabhängig von $\mathbb{1}_{\{N\leq n\}}$ für alle n, also insbesondere für $n = i-1$, ist, wonach X_i auch unabhängig von $\mathbb{1}_{\{N\geq i\}} = 1 - \mathbb{1}_{\{N\leq i-1\}}$ sein muss.
Nach Satz 1.6 gilt also $\mathbb{E}[X_i \cdot \mathbb{1}_{\{N\geq i\}}] = \mathbb{E}[X_i]\mathbb{E}[\mathbb{1}_{\{N\geq i\}}]$ und da die X_i identisch verteilt sind, stimmen auch deren Erwartungswerte überein:

$$\begin{aligned}
\mathbb{E}[S(N)] &= \sum_{i=1}^{\infty} \mathbb{E}[X_i]\mathbb{E}[\mathbb{1}_{\{N\geq i\}}] \\
&= \sum_{i=1}^{\infty} \mathbb{E}[X_1]\mathbb{P}\big(\{N \geq i\}\big) \\
&= \mathbb{E}[X_1]\mathbb{E}[N],
\end{aligned}$$

wobei letztere Gleichheit nach Lemma 1.4 gilt. □

1.3 Landau-Symbole

Um das asymptotische Verhalten von Zahlenfolgen und Funktionen zu beschreiben hat sich die Notation mithilfe der Landau-Symbole bewährt. In dieser Arbeit dienen sie zudem, wie auch in der Informatik üblich, als Maß für die Anzahl der Elementarschritte, die ein Algorithmus durchführt, in Abhängigkeit von der Größe des Inputs.

Sind $(a_n)_{n\in\mathbb{N}}, (b_n)_{n\in\mathbb{N}}$ Folgen reeller Zahlen, so schreiben wir

$$a_n = O(b_n) \text{ , falls } \limsup_{n\to\infty} \frac{|a_n|}{|b_n|} < \infty \text{ gilt.}$$

Ebenso definieren wir

$$a_n = o(b_n) \text{ , falls } \lim_{n\to\infty} \frac{|a_n|}{|b_n|} = 0 \text{ gilt.}$$

Sind weiter $f, g : \mathbb{N} \to \mathbb{R}$ Funktionen, so schreiben wir

$$g = O(f) \text{ (bzw. } g = o(f)),$$

wenn $a_n = O(b_n)$ (bzw. $a_n = o(b_n)$) für $a_n = g(n)$ und $b_n = f(n)$ für alle $n \in \mathbb{N}$ gilt.
Anschaulich bedeutet $g = O(f)$, dass g nicht wesentlich schneller als f wächst und $g = o(f)$, dass g langsamer als f wächst.
Ist sowohl $f = O(g)$ als auch $g = O(f)$, so schreiben wir

$$f = \Theta(g)$$

und sagen, dass f genauso schnell wächst wie g.

An dieser Stelle soll noch einmal betont werden, dass es sich lediglich um Schreibweisen und nicht um echte Gleichheiten handelt. Für ein leichteres Umgehen mit dieser Notation werden sich folgende Aussagen als hilfreich erweisen.

Lemma 1.9 *Sind $f, g, h : \mathbb{N} \to \mathbb{R}$ Funktionen und ist $f = O(h)$, sowie $g = O(h)$, so gilt*

(i) $f + g = O(h)$,

(ii) $O(g + h) = O(h)$ *und*

(iii) $c \cdot O(h) = O(h)$ *für eine Konstante* $c \in \mathbb{R}$.

Dabei sind (ii) und (iii) so zu lesen, dass für alle Funktionen $y : \mathbb{N} \to \mathbb{R}$, die jeweils die linke Seite erfüllen, auch die rechte gilt. Die entsprechenden Beweise ergeben sich direkt aus den Definitionen und der Dreiecksungleichung.

Beispiel 1.10 *Sei $n \in \mathbb{N}$ und $H(n) := \sum_{k=1}^{n} \frac{1}{k}$ die n-te harmonische Zahl. Da die Funktion $\frac{1}{x}$ monoton fallend ist, gilt einerseits*

$$ln(n) = \int_1^n \frac{1}{x} dx \leq \sum_{k=1}^{n} \frac{1}{k} \text{ und andererseits } \sum_{k=2}^{n} \frac{1}{k} \leq \int_1^n \frac{1}{x} dx = ln(n).$$

Durch diese beidseitige Approximation erhalten wir

$$ln(n) \leq H(n) \leq ln(n) + 1.$$

Folglich gilt $H(n) = ln(n) + \Theta(1)$ und damit auch $H(n) = ln(n) + O(1)$.

Damit sollten die für den nachfolgenden Teil vorbereitenden Grundlagen vollständig zusammengestellt sein und wir können mit der Analyse von *Bogosort* beginnen.

2 Bogosort

Bogosort, auch bekannt als Monkey- oder Stupidsort, ist ein randomisiertes Sortierverfahren, bei dem die zufällige Komponente eine besonders große Rolle spielt. Wir geben zunächst den Pseudocode an:

Algorithm 1: Bogosort

Input: array $a[1, ..., n]$
Output: $[1, ..., n]$
while $a[1, ..., n]$ *is not sorted* **do**
    randomly permute $a[1, ..., n]$
end

Das Überprüfen auf Sortiertheit und das zufällige Mischen der Liste funktioniert dabei folgendermaßen:

Procedure sorted

Input: array $a[1, ..., n]$
Output: *true*, if the array is sorted and $false$ otherwise
for $i = 1$ **to** $n - 1$ **do**
    if $a[i] > a[i + 1]$ **then**
        return *false*
    end
end
return *true*

Procedure randomly permute

Input: array $a[1, ..., n]$
Output: permuted array $a[1, ..., n]$
for $i = 1$ **to** $n - 1$ **do**
    $j := rand(i, ..., n)$
    $swap(a[i], a[j])$
end

Dabei liefert die Funktion $rand(i, ..., n)$ einen gleichverteilt zufälligen Wert aus $\{i, ..., n\}$ und *swap* vertauscht die entsprechenden Werte in der Liste.
Offensichtlich erhielt das Verfahren seine bezeichnenden Namen, da es lediglich testet, ob die gegebene Liste sortiert ist und sie dann gegebenenfalls neu mischt.
Als erste Beobachtung sei bemerkt, dass der Algorithmus keineswegs ter-

minieren muss. Betrachtet man allerdings Markovs Ungleichung (siehe Satz 1.7)

$$\mathbb{P}\big(\{X \geq t\}\big) \leq \frac{\mathbb{E}[X]}{t}$$

und identifiziert die Zufallsvariable X mit der Laufzeit des Algorithmus, so wird die Wahrscheinlichkeit, dass diese größer als jedes $t > 0$ ist, zunehmend gering, sofern ihr Erwartungswert nur endlich ist. Letzteres werden wir in diesem Kapitel zeigen.

2.1 Überprüfen auf Sortiertheit

Im Folgenden betrachten wir zunächst die Anzahl der Vergleiche, die im Test ob eine gegebene, gleichverteilt zufällig gewählte Liste bereits sortiert ist, benötigt wird. Wir gehen dabei davon aus, dass die Liste von links nach rechts durchlaufen wird und insgesamt $n \in \mathbb{N}$ Elemente enthält. Offenbar hängt die erwartete Anzahl der Vergleiche davon ab wie diese Elemente genau gewählt werden. In diesem Abschnitt werden insgesamt drei Fälle betrachtet.
Im ersten setzen wir voraus, dass jedes Element genau einmal auftritt. Dazu betrachten wir den Wahrscheinlichkeitsraum aller Permutationen, also die Gruppe S_n zusammen mit der Gleichverteilung.
Das Ergebnis wird dann auch im Hinblick auf das Verhalten für größer werdendes n untersucht, um es mit nachfolgenden Ergebnissen zu vergleichen, e bezeichnet dabei die Eulersche Zahl.

Satz 2.1 *Sei $n \in \mathbb{N}$ und $\omega \in \Omega_n = S_n$ eine Permutation in $\{1, ..., n\}$ und $C : \Omega_n \to \mathbb{R}$ die Zufallsvariable, die die Anzahl der Vergleiche im Test ob ω sortiert ist angibt. Dann gilt*

$$\mathbb{E}[C] = \sum_{i=1}^{n-1} \frac{1}{i!} \sim e - 1.$$

Beweis : Ist $n = 1$, so ist ω bereits sortiert und der Test benötigt keine Vergleiche: $\mathbb{E}[C] = 0 = \sum_{i=1}^{0} \frac{1}{i!}$.
Sei also $n \geq 2$ und für $1 \leq k < n$ sei I_k die Indikatorzufallsvariable des Ereignisses $\{$*die ersten k Elemente von ω sind sortiert*$\}$. Dann nimmt I_k genau dann den Wert 1 an, wenn $C(\omega) \geq k$ gilt, da der Algorithmus abbricht, sobald er einen Fehlstand findet.
Damit genügt es, die Anzahl der Permutationen zu bestimmen, deren ersten k Elemente sortiert sind. Das sind $\binom{n}{k} \cdot (n-k)!$, da wir die ersten k Elemente frei

wählen können und für die restlichen $n-k$ Elemente keinerlei Einschränkung vorliegt. Insgesamt gibt es $n!$ Permutationen in S_n und wir erhalten:

$$\mathbb{P}\big(\{C \geq k\}\big) = \mathbb{P}\big(\{I_k = 1\}\big) = \frac{\binom{n}{k} \cdot (n-k)!}{n!} = \frac{1}{k!}.$$

Da C positiv ist, schließlich muss mindestens ein Vergleich stattfinden, folgt für den Erwartungswert mit Lemma 1.4(ii) und der Definition von e

$$\mathbb{E}[C] = \sum_{k>0} \mathbb{P}\big(\{C \geq k\}\big) = \sum_{k>0} \mathbb{P}\big(\{I_k = 1\}\big) = \sum_{k=1}^{n-1} \frac{1}{k!} \sim e - 1.$$

□

An dieser Stelle sei bemerkt, dass der Algorithmus, egal wie groß n gewählt wird, im Mittel dennoch konstant viele Vergleiche, nämlich $e - 1 \approx 1.72$, benötigt, um zu testen ob die Liste sortiert ist. Im aufwändigsten Fall, die Liste ist sortiert, wären es immerhin $n - 1$ Vergleiche.

Korollar 2.2 *In der Situation von Satz 2.1 gilt*

$$\mathbb{E}[C] = e - 1 - O(\frac{1}{n!}).$$

Beweis : Nach Satz 2.1 gilt $\mathbb{E}[C] = \sum_{k=1}^{n-1} \frac{1}{k!}$.
Betrachtet man nun die Restgliedabschätzung der $(n-1)$-ten Taylorentwicklung der Exponentialfunktion an der Entwicklungsstelle 0 ausgewertet in $x = 1$

$$exp(1) - 1 = \sum_{k=1}^{n-1} \frac{1}{k!} + R_{n-1}(1;0) \leq \sum_{k=1}^{n-1} \frac{1}{k!} + O(\frac{1}{n!})$$

erhält man direkt

$$\mathbb{E}[C] \geq e - 1 - O(\frac{1}{n!}).$$

Die andere Richtung lässt sich per Induktion über n zeigen. Der Induktionsanfang ist dabei klar, im Induktionsschritt erhält man mit Lemma 1.9

$$\sum_{k=1}^{n-1} \frac{1}{k!} + \frac{1}{n!} \overset{\text{IV}}{\leq} e - 1 - O(\frac{1}{n!}) + \frac{1}{n!} \leq e - 1 - O(\frac{1}{(n+1)!}).$$

□

Etwas komplizierter wird es, wenn wir jedes Element in der Liste nicht nur einmal sondern mehrfach zulassen. Dazu betrachten wir wiederum zwei Fälle. Im ersten Fall wählen wir Elemente aus $\{1, ..., n\}$, im zweiten aus $\{0, 1\}$.
Für den ersten Fall wird folgende, hier erstmal zusammenhangslose, Bemerkung hilfreich sein.

Bemerkung 2.3 *Sei $n \geq 2$ und $k \in \{1, ..., n-1\}$. Wir wollen nun die Möglichkeiten zählen, gewisse ganzzahlige $x_i \geq 0, 1 \leq i \leq n$, so zu wählen, dass $\sum_{i=1}^{n} x_i = k$ gilt.*
Man stelle sich dazu k Steine getrennt durch $n-1$ Stöcke nebeneinander aufgereiht vor. Gilt nun $k = \sum_{i=1}^{n} x_i$, so gibt es n Gruppen mit jeweils $x_i \geq 0$, $1 \leq i \leq n$, Steinen. Zusammen ergibt das $n-k+1$ Plätze, von denen wir $n-1$ als Stöcke, bzw. k als Steine wählen. Das macht also $\binom{n-1+k}{k}$ Möglichkeiten.

Satz 2.4 *Sei $n \in \mathbb{N}$ beliebig und $\omega \in \Omega_n = \{1, ..., n\}^n$ gleichverteilt zufällig gegeben. Die Zufallsvariable $C : \Omega_n \to \mathbb{N}_0$ zähle die Vergleiche im Test, ob ω sortiert ist. Dann gilt*

$$\mathbb{E}[C] = \sum_{k=1}^{n-1} \binom{n-1+k}{k} \left(\frac{1}{n}\right)^k \sim e - 1.$$

Beweis : Ist $n = 1$, so ist ω offenbar bereits sortiert und C nimmt ausschließlich den Wert 0 an: $\mathbb{E}[C] = 0 = \sum_{k=1}^{0} \binom{k}{k} (\frac{1}{1})^k$.
Sei also $n \geq 2$. Dann nimmt C den Wert $1 \leq k \leq n-1$ an, wenn der Algorithmus nach k Vergleichen erkennt, dass ω nicht sortiert ist. Dann kann man ω folgendermaßen darstellen

$$\underbrace{1^{t_1} 2^{t_2} \cdots n^{t_n}}_{k} \underbrace{* \cdots *}_{n-k},$$

wobei die $t_i \geq 0$, $1 \leq i \leq n$, die Anzahl der nacheinander vorkommenden Zahlen aus $\{1, ..., n\}$ angeben und $\sum_{i=1}^{n} t_i = k$ gilt. Es genügt also, die Möglichkeiten zu zählen, sodass $k = \sum_{i=1}^{n} t_i$, für $t_i \geq 0$ gilt.
An dieser Stelle wird obige Bemerkung 2.3 nützlich und wir erhalten $\binom{n-1+k}{k}$ Möglichkeiten für die ersten k Elemente. Die restlichen, oben mit $*$ gekennzeichneten Elemente sind beliebig wählbar, dafür gibt es n^{n-k} Möglichkeiten, insgesamt also

$$\binom{n-1+k}{k} n^{n-k}.$$

In Ω_n gibt es insgesamt n^n Elemente, also berechnet sich die Wahrscheinlichkeit, dass C einen Wert größer oder gleich k annimmt wie folgt

$$\mathbb{P}\big(\{C \geq k\}\big) = \binom{n-1+k}{k} n^{n-k} \cdot \frac{1}{n^n} = \binom{n-1+k}{k} \left(\frac{1}{n}\right)^k.$$

Damit erhalten wir für den Erwartungswert von C

$$\mathbb{E}[C] = \sum_{k=1}^{n-1} \mathbb{P}\big(\{C \geq k\}\big) = \sum_{k=1}^{n-1} \binom{n-1+k}{k} \left(\frac{1}{n}\right)^k .$$

Mit $x := \frac{1}{n}$ folgt

$$\mathbb{E}[C] = \sum_{k=0}^{\infty} \binom{n-1+k}{k} x^k - \sum_{k=n}^{\infty} \binom{n-1+k}{k} x^k - 1. \tag{1}$$

Zunächst widmen wir uns der ersten Summe.
Da $\binom{n-1+k}{k}$ die Anzahl der Möglichkeiten n nichtnegative Elemente zu k zu summieren darstellt, folgt

$$\begin{aligned}
\sum_{k=0}^{\infty} \binom{n-1+k}{k} x^k &= \sum_{k=0}^{\infty} \sum_{\substack{a_1+\ldots+a_n=k \\ a_i \geq 0}} x^{a_1} \cdot \ldots \cdot x^{a_n} \\
&= (\sum_{a_1=0}^{\infty} x^{a_1}) \cdot \ldots \cdot (\sum_{a_n=0}^{\infty} x^{a_n}) \\
&= \left(\frac{1}{1-x}\right)^n .
\end{aligned}$$

Mit $x = \frac{1}{n}$ erhalten wir die Folge $(y_n)_{n>1} := (\frac{n}{n-1})^n$, deren Grenzwert wir nun betrachten wollen. Dass dieser existiert kann mit der Ungleichung vom harmonischen und geometrischen Mittel gezeigt werden. Diese besagt, dass für $x_i > 0$, $1 \leq i \leq m$,

$$\sqrt[m]{\prod_{i=1}^{m} x_i} \geq \frac{m}{\sum_{i=1}^{m} \frac{1}{x_i}}$$

gilt, und wir erhalten die Monotonie von y_n:

$$\sqrt[n+1]{(\frac{n}{n-1})^n \cdot 1} \geq \frac{n+1}{n}.$$

Außerdem ist y_n für $n > 1$ offenbar nach unten durch 1 beschränkt und der Grenzwert kann berechnet werden:

$$\begin{aligned}
\lim_{n\to\infty} y_n = \lim_{n\to\infty} y_{n+1} &= \lim_{n\to\infty} \left(\frac{n+1}{n}\right)^{n+1} \\
&= \lim_{n\to\infty} \left(\frac{n+1}{n}\right)^n \cdot \left(\frac{n+1}{n}\right) \\
&= e + \lim_{n\to\infty} \left(\frac{n+1}{n}\right)^n \cdot \frac{1}{n}.
\end{aligned}$$

Da $(\frac{n+1}{n})^n$ gegen e konvergiert, ist die Folge offenbar beschränkt und $(\frac{(n+1)^n}{n^{n+1}})_{n\geq 1}$ ist eine Nullfolge. Wir erhalten, dass auch $(y_n)_n$ gegen e konvergiert.
Damit können wir uns der zweiten Summe in (1) zuwenden.
Offenbar gilt $\binom{2n-1+k}{k+n} \leq 2^{2n-1+k}$ und damit

$$\begin{aligned}
\sum_{k=n}^{\infty} \binom{n-1+k}{k} x^k &= x^n \sum_{k=0}^{\infty} \binom{2n-1+k}{k+n} x^k \\
&\leq x^n \sum_{k=0}^{\infty} 2^{2n-1+k} x^k \\
&= 2^{2n-1} x^n \sum_{k=0}^{\infty} (2x)^k \\
&= 2^{2n-1} x^n \frac{1}{1-2x} \text{ , für } x < \frac{1}{2}.
\end{aligned} \tag{2}$$

Mit $x = \frac{1}{n}$ und $n > 2$ erhalten wir in (2) die nichtnegative Folge

$$(z_n)_{n>2} := (\frac{2^{2n-1}}{n^{n-1}(n-2)})_{n>2}$$

und für den Erwartungswert in (1) insgesamt

$$\left(\frac{n}{n-1}\right)^n - z_n - 1 \leq \mathbb{E}[C] \leq \left(\frac{n}{n-1}\right)^n - 1.$$

Für $n \geq 8$ gilt nun

$$z_n \leq \frac{2^{2n-1}}{n^{n-1}} = 2 \cdot \frac{2^{2n-2}}{n^{n-1}} = 2 \cdot \left(\frac{4}{n}\right)^{n-1} \leq 2 \cdot \left(\frac{4}{8}\right)^{n-1},$$

die Folge $(z_n)_{n>2}$ wird demnach durch eine Nullfolge majorisiert, sodass wir insgesamt

$$\mathbb{E}[C] \sim e - 1$$

erhalten. $\square$

Die mittlere Anzahl der Vergleiche stimmt also in den bisher betrachteten Fällen ab einer gewissen Listenlänge überein, während sie im nächsten, dem binären Fall vergleichsweise drastisch ansteigt.

Satz 2.5 *Sei $n \in \mathbb{N}$ beliebig und $\omega \in \Omega_{\{0,1\}} = \{0,1\}^n$ gleichverteilt zufällig gegeben. Die Zufallsvariable $C : \Omega_{\{0,1\}} \to \mathbb{N}_0$ zähle die Vergleiche, die im Test, ob ω sortiert ist, benötigt werden. Dann gilt*

$$\mathbb{E}[C] = 3 - \frac{n+2}{2^{n-1}} \sim 3.$$

Beweis : Ist $n = 1$, so ist ω offenbar bereits sortiert und es gilt
$\mathbb{E}[C] = 0 = 3 - \frac{3}{2^0}$. Bei $n = 2$ benötigt der Test genau einen Vergleich und wir erhalten $\mathbb{E}[C] = 1 = 3 - \frac{4}{2}$.
Sei nun $n \geq 3$ und $k \in \{1, ..., n-2\}$. Nimmt C den Wert k an, so erkennt der Algorithmus nach genau k Vergleichen, dass ω nicht sortiert ist und somit von folgender Form sein muss

$$\underbrace{0\cdots0}_{l}\underbrace{1\cdots1}_{k-l>0}0\underbrace{*\cdots*}_{n-k-1},$$

d.h. ω beginnt mit $l \geq 0$ Nullen, worauf $k - l$ Einsen, gefolgt von mindestens einer Null, folgen und die restlichen $n - k - 1$ Stellen beliebig sind. Dafür gibt es

$$\sum_{l=0}^{k-1} 2^{n-k-1} = k2^{n-k-1}$$

Möglichkeiten. Insgesamt gibt es 2^n n-Tupel mit Einträgen aus $\{0,1\}$. Damit ergibt sich für die Wahrscheinlichkeit, dass der Algorithmus k Vergleiche benötigt

$$\mathbb{P}\big(\{C = k\}\big) = \frac{k2^{n-k-1}}{2^n} = k \cdot \left(\frac{1}{2}\right)^{k+1}.$$

Es bleibt der Fall, dass C den Wert $n - 1$ annimmt. Dann ist ω entweder bereits sortiert oder von folgender Form

$$\underbrace{0\cdots0}_{l}\ \underbrace{1\cdots1}_{n-1-l>0}\ 0.$$

Dafür gibt es $\sum_{l=0}^{n-2} 1 = n - 1$ Möglichkeiten. Außerdem gibt es insgesamt $\sum_{l=0}^{n} 1 = n + 1$ sortierte n-Tupel, sodass sich die Wahrscheinlichkeit, dass C den Wert $n - 1$ annimmt, wie folgt berechnet

$$\mathbb{P}\big(\{C = n - 1\}\big) = \frac{n-1}{2^n} + \frac{n+1}{2^n} = \frac{n}{2^{n-1}}.$$

Insgesamt ergibt sich für den Erwartungswert von C

$$\begin{aligned}\mathbb{E}[C] &= \sum_{k=1}^{n-2} k\mathbb{P}\big(\{C = k\}\big) + (n-1)\mathbb{P}\big(\{C = n-1\}\big) \\ &= \sum_{k=1}^{n-2} k^2 \left(\frac{1}{2}\right)^{k+1} + (n-1) \cdot \frac{n}{2^{n-1}},\end{aligned}$$

und mit $x := \frac{1}{2}$

$$\mathbb{E}[C] = \frac{1}{2} \sum_{k=1}^{n-2} k^2 x^k + (n-1) \cdot \frac{n}{2^{n-1}}.$$

Wir suchen also einen geschlossenen Ausdruck von $S_m := \sum_{k=1}^{m} k^2 x^k$ mit $m := n-2$.
Zunächst erhalten wir durch Indexverschiebung

$$\begin{aligned}S_m + (m+1)^2 x^{m+1} = S_{m+1} &= \sum_{k=0}^{m} (k+1)^2 x^{k+1} \\ &= \sum_{k=0}^{m} k^2 x^{k+1} + \sum_{k=0}^{m} 2k x^{k+1} + \sum_{k=0}^{m} x^{k+1} \\ &= xS_m + 2x \sum_{k=0}^{m} k x^k + x \cdot \frac{1 - x^{m+1}}{1-x},\end{aligned}$$

wobei sich die letzte Summe aus der m-ten Partialsumme der geometrischen Reihe ergibt, und schließlich durch Äquivalenzumformung

$$S_m = \frac{2x}{1-x} \sum_{k=0}^{m} k x^k + \frac{x - (m+1)^2 x^{m+1} + ((m+1)^2 - 1) x^{m+2}}{(1-x)^2}. \tag{3}$$

Das Problem reduziert sich somit auf das Finden eines geschlossenen Ausdrucks für $T_m := \sum_{k=0}^{m} k x^k$. Dazu gehen wir analog vor:

$$\begin{aligned}T_m + (m+1) x^{m+1} = T_{m+1} &= \sum_{k=0}^{m} (k+1) x^{k+1} \\ &= \sum_{k=0}^{m} k x^{k+1} + \sum_{k=0}^{m} x^{k+1} \\ &= xT_m + x \cdot \frac{1 - x^{m+1}}{1-x}\end{aligned}$$

und erhalten durch Äquivalenzumformung

$$T_m = \sum_{k=0}^{m} kx^k = \frac{mx^{m+2} - (m+1)x^{m+1} + x}{(1-x)^2}. \tag{4}$$

Damit erhalten wir nach Einsetzen in (3) und weiterem Umformen

$$S_m = \frac{-m^2x^{m+3} + (2m^2 + 2m - 1)x^{m+2} - (m+1)^2x^{m+1} + x^2 + x}{(1-x)^3}.$$

Einsetzen von $x = \frac{1}{2}$ und $m = n - 2$ liefert schließlich

$$\mathbb{E}[C] = 3 - \frac{n+2}{2^{n-1}} \sim 3.$$

□

2.2 Anzahl der Vertauschungen

Bevor wir uns nun der im Algorithmus benötigten gesamten Anzahl an Vergleichen zuwenden, widmen wir uns zunächst der Anzahl an getätigten Vertauschungen. Ab jetzt betrachten wir allerdings nur noch den ersten Fall aus vorherigem Kapitel, die gegebene Liste ω sei stets eine gleichverteilt zufällig gegebene Permutation der Zahlen 1 bis n. Da sich der sortierte Fall stark vom unsortierten unterscheidet, betrachten wir diesen gesondert.

Satz 2.6 *Sei $\omega \in S_n$ die mit Bogosort zu sortierende Liste und $S : S_n \to \mathbb{N}_0$ die Zufallsvariable, die die Anzahl der Vertauschungen im Algorithmus angibt. Dann gilt*

$$\mathbb{E}[S] = \begin{cases} 0 & \text{, falls } \omega \text{ sortiert ist,} \\ (n-1)n! & \text{, sonst.} \end{cases}$$

Beweis : Da der Algorithmus zuerst testet, ob ω bereits sortiert ist, finden in diesem Fall keine Vertauschungen statt.
Wir betrachten also den Fall, dass ω nicht sortiert ist. Sei dazu $I : S_n \to \mathbb{N}_0$ die Zufallsvariable, die die Anzahl der Iterationen, die der Algorithmus benötigt bis ω sortiert ist, angibt. I zählt also die Fehlversuche bis zum ersten Erfolg, für dessen Eintreten die Wahrscheinlichkeit $p = \frac{1}{n!} \in (0,1)$ ist. Nach Beispiel 1.3(iii) ist I also geometrisch verteilt mit

$$\mathbb{P}(\{I = i\}) = (1-p)^i p = \left(1 - \frac{1}{n!}\right)^i \frac{1}{n!} \text{ für } i = 0, 1, 2, ...$$

und

$$\mathbb{E}[I] = \frac{1}{p} = n!.$$

Da nun in jedem Mischvorgang genau $n-1$ Vertauschungen vorgenommen werden, gilt für die Anzahl der Vertauschungen insgesamt

$$S = (n-1)I$$

und mit der Linearität des Erwartungswerts

$$\mathbb{E}[S] = \mathbb{E}[(n-1)I] = (n-1)n!.$$

□

2.3 Anzahl der Vergleiche

Im Folgenden wollen wir nun die erwartete Anzahl der insgesamt im Algorithmus durchgeführten Vergleiche bestimmen.
Dazu nehmen wir zusätzlich an, dass der Algorithmus nicht stoppt, sobald die Liste sortiert ist und assoziieren eine Folge $(C_i)_{i\geq 0}$ von Zufallsvariablen mit den Phasen des Überprüfens der Liste auf Sortiertheit. Dabei zählt C_i die getätigten Vergleiche vor dem Mischen in der i-ten Iteration.
Die Zufallsvariable I gebe weiterhin die Anzahl der Iterationen und die Zufallsvariable C ab sofort die Anzahl aller im Algorithmus durchgeführten Vergleiche an. Offenbar gilt dann

$$C = \sum_{i=0}^{I} C_i.$$

Um mit so einer Summe umzugehen, verwenden wir das in Kapitel 1.2 erwähnte Konzept von *Stoppzeiten* und stellen fest:

Die bereits eingeführte Zufallsvariable I der Anzahl der Iterationen von Bogosort ist eine Stoppzeit bezüglich obiger Folge von Zufallsvariablen $(C_i)_{i\geq 1}$, da der Erwartungswert $\mathbb{E}[I] = n!$ endlich ist und der Zeitpunkt, zu dem der Algorithmus stoppt, nicht von zukünftigen Ereignissen abhängt, die Zufallsvariable $\mathbb{1}_{\{I\leq n\}}$ also unabhängig von $(C_i)_{i>n}$ ist, für alle $n \in \mathbb{N}_0$.

Mithilfe der in Abschnitt 1.2, Satz 1.8 bewiesenen *Formel von Wald* können wir nun die gesamte im Algorithmus benötigte Anzahl an Vergleichen bestimmen.

Satz 2.7 *Die Zufallsvariable C zähle die Vergleiche in Bogosort für eine Liste ω der Länge n und $c(\omega)$ zähle die Vergleiche, die der Algorithmus benötigt um festzustellen, ob ω sortiert ist. Dann gilt*

$$\mathbb{E}[C] = \begin{cases} c(\omega) = n - 1 & \text{, falls } \omega \text{ sortiert ist,} \\ c(\omega) + (e-1)n! - O(1) & \text{, sonst.} \end{cases}$$

Beweis : Ist ω sortiert, muss der Algorithmus dies nur feststellen und stoppt nach $n-1$ Vergleichen.
Sei ω also nicht sortiert. Nun zählt C_0 die Vergleiche vor dem ersten Mischen, C_0 stimmt also auch in Wahrscheinlichkeit mit $c(\omega)$ überein, d.h. es gilt $\mathbb{P}\big(\{C_0 = c(\omega)\}\big) = 1$.
Mit der Linearität erhalten wir für den Erwartungswert

$$\begin{aligned} \mathbb{E}[C] &= \mathbb{E}[\sum_{i=0}^{I} C_i] = \mathbb{E}[C_0] + \mathbb{E}[\sum_{i=1}^{I} C_i] \\ &= c(\omega) + \mathbb{E}[\sum_{i=1}^{I} C_i]. \end{aligned}$$

Weiter sind die $(C_i)_{i\geq 1}$ unabhängig identisch verteilt und I ist eine Stoppzeit. Wir erhalten mit der *Formel von Wald*

$$\mathbb{E}[\sum_{i=1}^{I} C_i] = \mathbb{E}[C_1]\mathbb{E}[I]$$

und weiter mit Korollar 2.2

$$\mathbb{E}[\sum_{i=1}^{I} C_i] = (e - 1 - O(\frac{1}{n!}))n! = (e-1)n! - O(1).$$

$\square$

Zusammenfassend erhalten wir für die gesamte im Algorithmus benötigte Anzahl an Vergleichen erwartungsgemäß hohe aber immerhin endliche Werte.

Korollar 2.8 *Die Zufallsvariable C zähle wie oben die Vergleiche in Bogosort für eine Liste ω der Länge n. Dann gilt*

$$\mathbb{E}[C] = \begin{cases} n-1 & \text{, im besten Fall,} \\ (e-1)n! + n - O(1) & \text{, im schlechtesten Fall,} \\ (e-1)n! - O(1) & \text{, im Mittel.} \end{cases}$$

Beweis : Es müssen nur die entsprechenden Werte für $c(\omega)$ im vorigen Satz 2.7 eingesetzt werden. Im besten Fall ist die Liste bereits sortiert und der Algorithmus benötigt genau $n-1$ Vergleiche um dies festzustellen.
Im schlimmsten Fall ist ω unsortiert und der Algorithmus benötigt $n-1$ Vergleiche um dies festzustellen. Mit Satz 2.7 erhalten wir

$$\mathbb{E}[C] = n - 1 + (e-1)n! - O(1) = (e-1)n! + n - O(1).$$

Mit Satz 2.1 gilt für die Anzahl an benötigten Vergleichen im durchschnittlichen Fall $c(\omega) = e - 1 - O(\frac{1}{n!})$, also

$$\mathbb{E}[C] = (e-1)n! - O(1)$$

im Mittel. $\square$

3 Quicksort

Wie der Name schon vermuten lässt kommt der im Folgenden untersuchte Sortieralgorithmus in der Regel schneller zu einem Ergebnis als Bogosort. Ein weiterer Unterschied besteht darin, dass wir hier von einer Liste mit paarweise verschiedenen Einträgen und einer darauf definierten strengen Totalordnung ausgehen müssen. Hier zunächst der Pseudocode:

Algorithm 2: Quicksort

```
Input: array a[a_1, ..., a_n]
initialize empty arrays: less, greater
if n > 1 then
    select pivot x ∈ {a_1, ..., a_n}
    for y ∈ {a_1, ..., a_n} \ {x} do
        if y < x then
            add y to less
        end
        if y > x then
            add y to greater
        end
        Quicksort(less), Quicksort(greater)
    end
    return array a := concatenate(less, x, greater)
end
else
    return a
end
```

Die grundlegende Idee dieses sogenannten *Divide and Conquer* Verfahrens ist, die gegebene Liste in zwei kleinere Teillisten, hier *less* und *greater*, anhand einer Entscheidungsvariable, des sogenannten Pivotelements, aufzuspalten und diese zuerst zu sortieren. Dabei befinden sich alle Elemente, die kleiner als das gewählte Pivotelement sind, in der Liste *less* und alle die größer sind in *greater*.

Beim Wählen des Pivotelements gibt es natürlich mehrere Möglichkeiten. Wir wollen im Folgenden zwei Varianten untersuchen. In der ersten wird das Pivotelement gleichverteilt zufällig aus der gegebenen Liste gewählt, die zweite wählt immer das erste in der Liste vorkommende Element unter Voraussetzung einer gleichverteilt zufällig gewählten Liste im Input. Einmal findet sich also die Randomisierung im Algorithmus wieder und einmal ausschließlich in der probabilistischen Analyse.

Zunächst wollen wir aber die Extremfälle betrachten. Da keine direkten Vertauschungen stattfinden wird die Laufzeit hier ausschließlich anhand der benötigten Vergleiche bestimmt.
Im *worst case* werden die Vergleiche in der Summe maximal, wenn eine der Unterlisten *less* oder *greater* leer ist. Das passiert, wenn das gewählte Pivotelement gerade das kleinste oder größte in der gegebenen Liste ist. Dann werden im ersten Durchlauf $n-1$, im zweiten $n-2$, ..., und im letzten nur noch ein Vergleich benötigt. Aufsummiert ergibt das

$$(n-1)+(n-2)+...+1=\frac{n(n-1)}{2}$$

Vergleiche im *worst case* und somit eine Laufzeit der Größenordnung $O(n^2)$.

Glücklicherweise zeigen *best* und *average case* ein besseres Laufzeitverhalten. Um die Anzahl der Vergleiche möglichst gering zu halten sollte die Liste in jedem Schritt etwa halbiert werden. Dann hat jede Teilliste im ersten Durchgang maximal $\frac{n}{2}$, im zweiten maximal $\frac{n}{4}$ und im k-ten Durchgang maximal $\frac{n}{2^k}$ Elemente. Im letzten Durchgang beinhalten die Teillisten zudem nur noch höchstens ein Element. Mit

$$\frac{n}{2^k}=1 \Leftrightarrow k=log_2(n)$$

erhalten wir, dass der Algorithmus höchstens $log_2(n)$ Durchläufe mit jeweils höchstens n Vergleichen benötigt. Im *best case* bewegt sich die Laufzeit also in der Größenordnung $O(n\ log_2(n))$.
Wie sich gleich zeigen wird entspricht dies auch der mittleren Laufzeit des randomisierten und auch der des deterministischen Algorithmus.

Satz 3.1 *Die Eingabeliste $S:=\{s_1,...,s_n\}$ sei fixiert und das Pivotelement werde gleichverteilt zufällig gewählt. Dann entspricht die erwartete Anzahl an Vergleichen in Quicksort $2n\ ln(n)+O(n)$ $(=O(n\ log_2(n))$.*

Beweis : Sei $\{t_1,...,t_n\}=S$ die sortierte Liste, d.h. $t_1<...<t_n$. Für $i<j$ sei X_{ij} die Indikatorzufallsvariable des Ereignisses $\{t_i$ *und* t_j *werden im Laufe des Algorithmus miteinander verglichen*$\}$. Dann entspricht die gesamte Anzahl an Vergleichen

$$X:=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X_{ij}$$

und somit die erwartete Anzahl an Vergleichen

$$\mathbb{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E}[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{P}\big(\{X_{ij} = 1\}\big).$$

Weiter werden t_i und t_j nur dann miteinander verglichen, wenn eins der beiden als Pivotelement gewählt wird und sich die beiden Elemente in der gleichen Liste befinden. Das ist genau dann der Fall, wenn t_i oder t_j als Pivotelement aus $T_{i,j} := \{t_i, t_{i+1}, ..., t_j\}$ gewählt wird, da sich, falls das Pivotelement x aus $S \setminus T_{i,j}$ gewählt wird, t_i und t_j weiterhin in derselben Unterliste befinden, ein Vergleich aber unmöglich wird, sobald x aus $T_{i,j} \setminus \{t_i, t_j\}$ gewählt wird. Da das Pivotelement gleichverteilt zufällig bestimmt wird erhalten wir

$$\mathbb{P}\big(\{X_{ij} = 1\}\big) = \frac{2}{j - i + 1}.$$

Mit $k := j - i + 1$ berechnet sich der Erwartungswert zu

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{k} = \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \\
&= \sum_{k=2}^{n} \sum_{i=1}^{n+1-k} \frac{2}{k} = \sum_{k=2}^{n} (n + 1 - k) \frac{2}{k} \\
&= \left((n+1) \sum_{k=2}^{n} \frac{2}{k} \right) - 2(n-1) \\
&= (2n+2) \sum_{k=1}^{n} \frac{1}{k} - 4n.
\end{aligned}$$

In Beispiel 1.10 wurde gezeigt, dass $\sum_{k=1}^{n} \frac{1}{k} = ln(n) + O(1)$ gilt, damit folgt die Behauptung. $\square$

Wird nun das Pivotelement nicht mehr zufällig gewählt, erhalten wir dasselbe Ergebnis, sofern wir von einer gleichverteilt zufällig gegebenen Liste ausgehen:

Satz 3.2 *Sei* $S := \{s_1, ..., s_n\}$ *eine gleichverteilt zufällig gewählte Liste von paarweise verschiedenen Elementen. Wird immer das erste in der Liste vorkommende Element als Pivotelement gewählt, so entspricht die erwartete Anzahl an in Quicksort getätigten Vergleichen* $2n\ ln(n) + O(1)$.

Beweis : Der Beweis kann völlig analog zu obigem geführt werden. Wieder können die zwei Elemente t_i und t_j nur miteinander verglichen werden, wenn eins davon aus $T_{i,j} = \{t_i, ..., t_j\}$ als Pivotelement gewählt wird. Die Wahrscheinlichkeit, dass eins der beiden Elemente an erster Stelle einer Liste steht ist aber wieder $\frac{2}{j-i+1}$, da wir von gleichverteilten Elementen in der Liste ausgehen, und die Behauptung folgt mit obiger Rechnung. $\square$

Obwohl die Ergebnisse der beiden vorgestellten Varianten von Quicksort übereinstimmen gibt es einen entscheidenden Unterschied, der das randomisierte Verfahren für die Anwendung attraktiver macht. Hier wird die betrachtete Laufzeit eine zufällige Größe, die nicht mehr von den konkreten Eingaben abhängig ist. Dagegen musste bei der Analyse der deterministischen Variante eine gleichverteilt zufällig gewählte Eingabe vorausgesetzt werden, was in konkreten Anwendungen unrealistisch sein kann.

4 Bubblesort

Die Analyse der bisher betrachteten Sortieralgorithmen beruhte in der Regel auf der Annahme eines gleichverteilt zufällig gegebenen Inputs. Wir wollen nun untersuchen, was passiert, wenn eine andere Wahrscheinlichkeitsverteilung vorliegt und ziehen dazu die in Beispiel 1.2 (iii) eingeführte Plackett-Luce-Verteilung heran.
Zum Einstieg gehen wir zunächst noch einen Schritt zurück und vergleichen die Ergebnisse der probabilistischen Analysen eines nicht randomisierten Sortieralgorithmus. Aufgrund seiner vergleichsweise simplen Analyse bietet sich dafür *Bubblesort*, hier nur auf Permutationen angewendet, deren Einträge trivialerweise einer strengen Totalordnung unterliegen, an.

Algorithm 3: Bubblesort

Input: permutation $\sigma = \sigma(1) \cdots \sigma(n) \in S_n$
Output: $id = 1 \cdots n \in S_n$
for $m = n$ **downto** 2 **do**
  for $i = 1$ **to** $m - 1$ **do**
    if $\sigma(i) > \sigma(i+1)$ **then**
      $swap(\sigma(i), \sigma(i+1))$
    end
  end
end

Der Algorithmus durchläuft demnach die als Wort gegebene Permutation von links nach rechts und vertauscht gegebenenfalls jeweils zwei Einträge. Die äußere Schleife verschiebt nach jedem Durchgang die rechte Schranke nach links, da nach jedem Durchlauf der inneren Schleife das jeweils größte Element des bisher unsortierten Teils der Permutation ganz hinten steht.
Die Anzahl der in dieser Version des Algorithmus benötigten Vergleiche ist schnell bestimmt:
In der inneren Schleife werden jeweils $m - 1$ Vergleiche durchgeführt, wobei m die Werte aus $\{2, ..., n\}$ annimmt. Das ergibt

$$(n-1) + (n-2) + ... + 1 = \frac{n(n-1)}{2},$$

Vergleiche insgesamt.
Daher widmen wir uns im Folgenden ausschließlich der Anzahl der benötigten Vertauschungen. Wie der nächste Satz zeigen wird, entspricht diese der Anzahl der vorhandenen Inversionen in der gegebenen Permutation.

Satz 4.1 *Bei einer Eingabe von $\sigma \in S_n$ führt Bubblesort genau $\#Inv(\sigma)$ Vertauschungen durch, bis σ sortiert ist.*

Beweis : Im Algorithmus findet genau dann eine Vertauschung statt, wenn $\sigma(i) > \sigma(i+1)$ für ein $1 \leq i \leq n-1$ ist. Sei also $(i, i+1)$ eine Inversion und

$$\sigma' := \sigma(1) \cdots \sigma(i-1)\sigma(i+1)\sigma(i)\sigma(i+2) \cdots \sigma(n)$$

die Permutation, nachdem die entsprechenden Einträge getauscht wurden. Wir zeigen

$$\#Inv(\sigma') = \#Inv(\sigma) - 1,$$

denn dann folgt wegen $\#Inv(id) = 0$ sofort die Behauptung.
Sei nun $1 \leq k < l \leq n$. Ist zudem $\{k, l\} \cap \{i, i+1\} = \emptyset$, so ist $(k, l) \in Inv(\sigma)$ genau dann, wenn $(k, l) \in Inv(\sigma')$, da die Permutation außerhalb der Indizes i und $i+1$ unverändert bleibt.
Ist weiter $l < i$, dann ist (l, i) eine Inversion in σ genau dann, wenn $(l, i+1)$ eine Inversion in σ' ist, und analog ist $(l, i+1)$ genau dann eine Inversion in σ, wenn $(l, i) \in Inv(\sigma')$ ist.
Für die andere Seite erhalten wir analog:
Ist $k > i+1$, so ist (i, k) eine Inversion in σ genau dann, wenn $(i+1, k) \in Inv(\sigma')$ ist, und $(i+1, k)$ ist eine Inversion in σ genau dann, wenn $(i, k) \in Inv(\sigma')$ gilt.
Außer dass die betrachtete Inversion $(i, i+1)$ durch die Vertauschung verschwindet, ändert sich also nichts an der Anzahl der Inversionen in σ'. □

4.1 mit der Gleichverteilung

Da die Anzahl der benötigten Vertauschungen mit der Anzahl der Inversionen in der gegebenen Permutation übereinstimmt, reicht es, diese zu bestimmen und wir erhalten folgenden Satz für den gleichverteilten Fall.

Satz 4.2 *Sei $\sigma \in S_n$ eine zufällig gleichverteilt gegebene Permutation und $X_G : S_n \to \mathbb{N}_0$ zähle die Vertauschungen, die Bubblesort durchführt um σ zu sortieren. Dann gilt*

$$\mathbb{E}[X_G] = \frac{n(n-1)}{4}.$$

Beweis : Ist $n = 1$, so ist σ bereits sortiert und es gilt $\mathbb{E}[X_G] = 0$.
Sei nun $n \geq 2$ und für $1 \leq i < j \leq n$ sei $X_{ij} : S_n \to \{0, 1\}$ die Indikatorzufallsvariable für das Ereignis $\{\sigma \in S_n \mid \sigma(i) > \sigma(j)\}$. Nach Satz 4.1 ist nun

$$\mathbb{E}[X_G] = \mathbb{E}[\sum_{i<j} X_{ij}] = \sum_{i<j} \mathbb{E}[X_{ij}] = \sum_{i<j} \mathbb{P}\big(\{X_{ij} = 1\}\big),$$

und da die zugrundeliegende Totalordnung streng ist und σ gleichverteilt zufällig gewählt wurde gilt schließlich

$$\sum_{i<j} \mathbb{P}\big(\{X_{ij} = 1\}\big) = \sum_{i<j} \frac{1}{2} = \binom{n}{2} \cdot \frac{1}{2} = \frac{n(n-1)}{4}.$$

$\square$

4.2 mit der Plackett-Luce-Verteilung

Wir wollen nun weiterhin die Anzahl der Vertauschungen in Bubblesort betrachten, diesmal unter der Annahme einer mittels der in Beispiel 1.2(iii) eingeführten Plackett-Luce-Verteilung gegebenen Permutation.
Der zugrundeliegende Wahrscheinlichkeitsraum wird demnach ab jetzt $(S_n, \mathbb{P}_x)$ für reelle, positive $x_k := x^k, 1 \leq k \leq n$, und

$$\mathbb{P}_x(\sigma) = \prod_{k=1}^{n} \frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{\sigma(n)}}$$

sein. Wegen Satz 4.1 gilt dann für die Zufallsvariable $X_{PL} : S_n \to \mathbb{N}_0$, die die Vertauschungen in Bubblesort zählt

$$X_{PL}(\sigma) = \#Inv(\sigma)$$

und $\mathbb{E}[X_{PL}]$ lässt sich als gebrochen rationale Funktion in x darstellen

$$\mathbb{E}[X_{PL}] = \sum_{\sigma \in S_n} \#Inv(\sigma) \prod_{k=1}^{n} \frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{\sigma(n)}}$$

$$= \frac{a_r x^r + ... + a_1 x + a_0}{b_s x^s + ... + b_1 x + b_0}, \tag{5}$$

wobei die $a_i, i = 0, ..., r$, und die $b_j, j = 0, ..., s$, so gewählt seien, dass der größte gemeinsame Teiler des Polynoms im Zähler und des Polynoms im Nenner von (5) im Ring der reellen Polynome gleich 1 ist, der Bruch also nicht weiter gekürzt werden kann und die Darstellung mit $b_s = 1$ eindeutig wird.

Beispiel 4.3 *Für $n = 3$ ergibt sich*

$$\mathbb{E}[X_{PL}] = \sum_{\sigma \in S_3} \#Inv(\sigma) \prod_{k=1}^{3} \frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{\sigma(3)}}$$

$$\begin{aligned}
&= 3\left(\frac{x^3}{x^3+x^2+x} \cdot \frac{x^2}{x^2+x}\right) + 2\left(\frac{x^3}{x^3+x+x^2} \cdot \frac{x}{x+x^2}\right) \\
&\quad + 2\left(\frac{x^2}{x^2+x^3+x} \cdot \frac{x^3}{x^3+x}\right) + \left(\frac{x^2}{x^2+x+x^3} \cdot \frac{x}{x+x^3}\right) \\
&\quad + \left(\frac{x}{x+x^3+x^2} \cdot \frac{x^3}{x^3+x^2}\right)
\end{aligned}$$

$$\begin{aligned}
&= 3\left(\frac{x^2}{x^2+x+1} \cdot \frac{x}{x+1}\right) + 2\left(\frac{x^2}{x^2+1+x} \cdot \frac{1}{1+x}\right) \\
&\quad + 2\left(\frac{x}{x+x^2+1} \cdot \frac{x^2}{x^2+1}\right) + \left(\frac{x}{x+1+x^2} \cdot \frac{1}{1+x^2}\right) \\
&\quad + \left(\frac{1}{1+x^2+x} \cdot \frac{x}{x+1}\right)
\end{aligned}$$

$$= \frac{(3x^3+2x^2+x)(x^2+1) + (2x^3+x)(x+1)}{(x^2+x+1)(x^2+1)(x+1)}$$

$$= \frac{3x^5+4x^4+6x^3+3x^2+2x}{(x^2+x+1)(x^2+1)(x+1)} = \frac{3x^3+x^2+2x}{(x^2+1)(x+1)}.$$

Mithilfe des Computeralgebrasystems Maple[1] berechnen wir weitere Werte für $n \le 5$:

$n = 1:$

$$0$$

$n = 2:$

$$\frac{x}{x+1}$$

$n = 3:$

$$\frac{3\,x^3+x^2+2x}{(x^2+1)(x+1)}$$

[1] Quellcode im Anhang

$n = 4:$

$$\frac{6\,x^6 + 3\,x^5 + 4\,x^4 + 6\,x^3 + 2\,x^2 + 3x}{(x^3+1)(x^2+1)(x+1)}$$

$n = 5:$

$$\frac{10\,x^{10} + 6\,x^9 + 7\,x^8 + 11\,x^7 + 13\,x^6 + 10\,x^5 + 7\,x^4 + 9\,x^3 + 3\,x^2 + 4x}{(x^4+1)(x^3+1)(x^2+1)(x+1)}$$

Als erste Beobachtung stellen wir fest:

Satz 4.4 *In der Darstellung (5) gilt*

(i) $a_0 = 0$ *und*

(ii) $r \geq \frac{1}{2}(n-1)n.$

Beweis : Für $n = 1$ lassen sich die Behauptungen schnell verifizieren. Sei also $n \geq 2$.

(i) Für $1 \leq k \leq n-1$ gibt es ein $m_k := \min\{\sigma(l) | k \leq l \leq n\}$, sodass
$$\frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{m_k} + ... + x^{\sigma(n)}} = \frac{x^{\sigma(k)-m_k}}{x^{\sigma(k)-m_k} + ... + 1 + ... + x^{\sigma(n)-m_k}} \tag{6}$$
mit $m_k \leq \sigma(k)$ gekürzt werden kann. Weiter können wir ohne Einschränkung annehmen, dass $\sigma \neq id$ gilt, da $\#Inv(id) = 0$ ist. Daher ist $m_k \leq \sigma(j) < \sigma(k)$ für $(k, j) \in Inv(\sigma)$, wir können somit in jedem Summanden von $\mathbb{E}[X_{PL}]$ mindestens ein x ausklammern und erhalten $a_0 = 0$.

(ii) Wegen (6) gilt für das Produkt in $\mathbb{E}[X_{PL}]$
$$\prod_{k=1}^{n} \frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{\sigma(n)}} = \prod_{k=1}^{n-1} \frac{x^{\sigma(k)-m_k}}{x^{\sigma(k)-m_k} + ... + 1 + ... + x^{\sigma(n)-m_k}}.$$
Der Exponent im Zähler wird somit maximal, falls $m_k = 1$ für alle k, also für $\sigma(n) = 1$ ist und entspricht in der Summe mindestens
$$r \geq \sum_{k=1}^{n-1} (\sigma(k) - 1) = \sum_{k=2}^{n} (k-1) = \frac{1}{2}(n-1)n.$$

$\square$

An dieser Stelle sei bemerkt, dass die Vermutung nahe liegt, dass sogar $r = \frac{1}{2}(n-1)n$ gilt, wir widmen uns allerdings zunächst dem Koeffizienten a_1. Bereits hier wird die Analyse etwas aufwändiger und wir benötigen folgendes Lemma.

Lemma 4.5 *Für $\sigma \in S_n$ gilt*

$$\#Inv(\sigma) \leq \sum_{k=1}^{n-1} \Big(\sigma(k) - \min\{\sigma(l)|k \leq l \leq n\}\Big).$$

Beweis : Wir beweisen die Behauptung mittels Induktion über $n \in \mathbb{N}$. Der Induktionsanfang, $n = 1$, ist klar, da $\#Inv(id) = 0$ gilt.
Sei nun $\pi \in S_n$ und $\sigma \in S_{n+1}$. Wir unterscheiden zwei Fälle:

1.Fall: In Wortschreibweise habe σ die Form $\sigma = \pi(1)\cdots\pi(n)(n+1)$.
Dann ist $\#Inv(\sigma) = \#Inv(\pi)$ und nach Induktionsvorraussetzung gilt

$$\begin{aligned}
\#Inv(\sigma) &\leq \sum_{k=1}^{n-1} \Big(\pi(k) - \min\{\pi(l)|k \leq l \leq n\}\Big) \\
&= \sum_{k=1}^{n-1} \Big(\sigma(k) - \min\Big\{\{\sigma(l)|k \leq l \leq n\} \cup \{n+1\}\Big\}\Big) \\
&= \sum_{k=1}^{n} \Big(\sigma(k) - \min\{\sigma(l)|k \leq l \leq n+1\}\Big),
\end{aligned}$$

da $\sigma(n) - \min\{\sigma(n), \sigma(n+1)\} = 0$ ist.

2.Fall: Für $2 \leq i \leq n$ habe σ eine der Formen
$\sigma = \pi(1)\cdots\pi(i-1)(n+1)\pi(i)\cdots\pi(n)$ oder $\sigma = (n+1)\pi(1)\cdots\pi(n)$.
Dann ist $\#Inv(\sigma) = \#Inv(\pi) + n + 1 - i$ und nach Induktionsvorraussetzung gilt

$$\begin{aligned}
\#Inv(\sigma) &\leq \sum_{k=1}^{n-1} \Big(\pi(k) - \min\{\pi(l)|k \leq l \leq n\}\Big) + n + 1 - i \\
&= \sum_{\substack{k=1\\k\neq i}}^{n} \Big(\sigma(k) - \min\{\sigma(l)|k \leq l \leq n+1\}\Big) + n + 1 - i \\
&\leq \sum_{\substack{k=1\\k\neq i}}^{n} \Big(\sigma(k) - \min\{\sigma(l)|k \leq l \leq n+1\}\Big) + n + 1 - \min\{\sigma(l)|i \leq l \leq n+1\},
\end{aligned}$$

da $i \geq \min\{\sigma(l)|i+1 \leq l \leq n+1\} = \min\{\sigma(l)|i \leq l \leq n+1\}$ ist. Angenommen dem wäre nicht so, also $\sigma(j) > i$ für alle $j \in \{i+1, ..., n+1\}$. Dann muss aber $\sigma(j) \leq i$ für alle $j \in \{1, ..., i\}$ sein und $\sigma(i) = n+1$ liefert den Widerspruch.

$\square$

Widmen wir uns nun dem Koeffizienten von x^1.

Satz 4.6 *Wird $\mathbb{E}[X_{PL}]$ wie in (5) als gebrochen rationale Funktion*

$$\mathbb{E}[X_{PL}] = \frac{a_r x^r + ... + a_1 x + a_0}{b_s x^s + ... + b_1 x + b_0}$$

dargestellt, so gilt

$$a_1 = n - 1.$$

Beweis : Zunächst kürzen wir wie in (6) jeden Faktor und erweitern anschließend auf einen gemeinsamen Nenner. Sei dazu

$$N(x) := \prod_{\substack{1 \le m \le n-1 \\ 1 \le k_1 < ... < k_m \le n-1}} \left(1 + x^{k_1} + ... + x^{k_m}\right)$$

der gemeinsame, von σ unabhängige Nenner und

$$E_\sigma(x) := \prod_{\substack{1 \le m \le n-1 \\ 1 \le k_1 < ... < k_m \le n-1 \\ \nexists k : \{0, k_1, ..., k_m\} = \{\sigma(k) - m_k, ..., \sigma(n) - m_k\}}} \left(1 + x^{k_1} + ... + x^{k_m}\right)$$

das Produkt der Faktoren, mit denen erweitert wird. Dabei ist wie in (6) $m_k = \min\{\sigma(l) | k \le l \le n\}$. Es folgt

$$\begin{aligned} \mathbb{E}[X_{PL}] &= \sum_{\sigma \in S_n} \#Inv(\sigma) \prod_{k=1}^{n-1} \frac{x^{\sigma(k) - m_k}}{x^{\sigma(k) - m_k} + ... + 1 + ... + x^{\sigma(n) - m_k}} \\ &= \sum_{\sigma \in S_n} \#Inv(\sigma) \frac{\prod_{k=1}^{n-1} x^{\sigma(k) - m_k} \cdot E_\sigma(x)}{N(x)} \\ &= \sum_{\sigma \in S_n} \#Inv(\sigma) x^{\sum_{k=1}^{n-1} (\sigma(k) - m_k)} \cdot \frac{E_\sigma(x)}{N(x)}. \end{aligned} \tag{7}$$

Nun gilt wegen Lemma 4.5

$$\sum_{k=1}^{n-1} (\sigma(k) - m_k) = \#Inv(\sigma) + \epsilon(\sigma)$$

für einen gewissen, insbesondere nichtnegativen Fehler $\epsilon(\sigma)$ und wir erhalten

$$\mathbb{E}[X_{PL}] = \frac{\sum_{\sigma \in S_n} \#Inv(\sigma) x^{\#Inv(\sigma)} \cdot x^{\epsilon(\sigma)} \cdot E_\sigma(x)}{N(x)}.$$

In dieser Darstellung wird deutlich, dass nun nur noch mit Termen der Form der Faktoren des Nenners $N(x)$ gekürzt werden kann. Diese Terme sind wiederum normierte Polynome mit absolutem Glied 1. Wegen Satz 4.4(i) beeinflusst eventuelles Kürzen mit solchen Polynomen den Koeffizienten von x^1 nicht. Da die Faktoren des Terms $E_\sigma(x)$, mit denen erweitert wurde, die gleiche Form haben, wir ohne Einschränkung $\#Inv(\sigma) > 0$ annehmen können und der Fehler $\epsilon(\sigma)$ nichtnegativ ist, erhalten wir für den Koeffizienten a_1 von x^1 mit Beispiel 1.1(iii)

$$a_1 = \#\{\sigma \in S_n | \#Inv(\sigma) = 1\} = n - 1.$$

□

Vergleichen wir unsere in (7) gefundene Darstellung von $\mathbb{E}[X_{PL}]$ mit den bereits berechneten Ergebnissen am Anfang dieses Abschnitts, so liegt die Vermutung nahe, dass sich alle Faktoren von $N(x)$ mit mehr als zwei Summanden restlos kürzen lassen. Formal hieße das, es gibt ein Polynom $f(x)$, sodass

$$\sum_{\sigma \in S_n} \#Inv(\sigma) x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)} E_\sigma(x) = f(x) \prod_{\substack{2 \leq m \leq n-1 \\ 1 \leq k_1 < \ldots < k_m \leq n-1}} \left(1 + x^{k_1} + \ldots + x^{k_m}\right)$$

gilt. Da der Leitkoeffizient des Polynoms im Zähler von $\mathbb{E}[X_{PL}]$ nicht durch eventuelles Kürzen von Termen der Form der Faktoren von $N(x)$ beeinflusst wird, schließlich sind diese Terme allesamt normierte Polynome, können wir diesen noch exakt bestimmen.

Satz 4.7 *In der Darstellung (5) gilt für den Leitkoeffizienten des Zählerpolynoms*

$$a_r = \frac{1}{2}(n-1)n.$$

Beweis : Betrachten wir die Darstellung von $\mathbb{E}[X_{PL}]$ in (7)

$$\mathbb{E}[X_{PL}] = \sum_{\sigma \in S_n} \#Inv(\sigma) x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)} \cdot \frac{E_\sigma(x)}{N(x)},$$

so wird deutlich, dass der Exponent von x im Zähler maximal wird für einerseits $\sum_{k=1}^{n-1}(\sigma(k) - m_k)$ maximal und für einen maximalen Exponenten von

x in $E_\sigma(x)$ andererseits. Dabei wird der Exponent in $E_\sigma(x)$ maximal, sobald er im Nenner von (6)

$$N_\sigma(x) := \prod_{k=1}^{n-1}(x^{\sigma(k)-m_k} + ... + 1 + ... + x^{\sigma(n)-m_k})$$

minimal wird. Schließlich gilt $E_\sigma(x) = N(x)/N_\sigma(x)$.
Nun wird der Exponent von x in $N_\sigma(x)$ minimal, falls

$$N_\sigma(x) = (x^{n-1} + ... + x + 1) \cdot ... \cdot (x^2 + x + 1)(x + 1) \tag{8}$$

gilt. Da außerdem $\sum_{k=1}^{n-1}(\sigma(k) - m_k)$ maximal sein soll, muss $m_k = 1$ für alle k, also $\sigma(n) = 1$ gelten. Diese Bedingung trifft zusammen mit (8) nur auf $\sigma = n(n-1)\cdots 1$ zu und für den Leitkoeffizienten des Zählerpolynoms ergibt sich mithilfe von Beispiel 1.1(ii)

$$a_r = \#Inv(n(n-1)\cdots 1) = \tfrac{1}{2}(n-1)n.$$

$\square$

Zusammenfassend konnten wir in diesem Abschnitt eine neue Darstellung der erwarteten Anzahl an Vertauschungen

$$\begin{aligned}
\mathbb{E}[X_{PL}] &= \sum_{\sigma \in S_n} \#Inv(\sigma) \prod_{k=1}^{n} \frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{\sigma(n)}} \\
&= \frac{a_r x^r + ... + a_1 x + a_0}{b_s x^s + ... + b_1 x + b_0} \\
&= \sum_{\sigma \in S_n} \#Inv(\sigma) x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)} \cdot \frac{E_\sigma(x)}{N(x)}
\end{aligned}$$

finden, sowie die Werte von a_0, a_1, a_r exakt bestimmen und eine Vermutung für die $b_0, ..., b_s$ aufstellen.

5 Bogosort mit Plackett-Luce

In diesem Abschnitt soll nun der in Kapitel 2 untersuchte Algorithmus mittels der in Beispiel 1.2(iii) eingeführten und im letzten Kapitel bereits verwendeten Plackett-Luce-Verteilung analysiert werden. Genauer wird hier, wie auch im vorigen Kapitel, nicht mehr von einer gleichverteilt zufällig gegebenen Liste, sondern von einer mittels Plackett-Luce-Verteilung gegebenen Permutation ausgegangen, der zugrundeliegende Wahrscheinlichkeitsraum ist also nach wie vor $(S_n, \mathbb{P}_x)$ für ein reelles $x > 0$.

5.1 Anzahl der Vertauschungen

Um die Anzahl der Vertauschungen, die Bogosort bei einer mittels Plackett-Luce-Verteilung gegebenen Permutation durchführt, zu bestimmen, gehen wir analog zum Beweis von Satz 2.6 vor und erhalten folgenden Satz.

Satz 5.1 *Sei $\omega \in S_n$ die mit Bogosort zu sortierende Liste und $S_{PL} : S_n \to \mathbb{N}_0$ die Zufallsvariable, die die Anzahl der Vertauschungen im Algorithmus angibt. Dann gilt*

$$\mathbb{E}[S_{PL}] = \begin{cases} 0 & \text{, falls } \omega \text{ sortiert ist,} \\ (n-1)\prod_{r=1}^{n-1}\sum_{s=0}^{n-r} x^s & \text{, sonst.} \end{cases}$$

Beweis : Ist die gegebene Liste bereits sortiert, so werden keine Vertauschungen durchgeführt. Sei $\omega \in S_n$ also nicht sortiert und $I_{PL} : S_n \to \mathbb{N}_0$ zähle die Iterationen, die Bogosort benötigt bis ω sortiert ist. Mit einer Erfolgswahrscheinlichkeit von

$$p := \mathbb{P}_x(id) = \prod_{r=1}^{n} \frac{x^r}{x^r + ... + x^n} = \prod_{r=1}^{n-1} \frac{1}{1 + x + ... + x^{n-r}} \in (0,1)$$

ist I_{PL} geometrisch verteilt und wir erhalten mithilfe von Beispiel 1.5(ii):

$$\mathbb{E}[I_{PL}] = \frac{1}{p} = \prod_{r=1}^{n-1}(1 + x + ... + x^{n-r}).$$

Weiterhin werden in jedem Mischvorgang $n-1$ Vertauschungen vorgenommen und wir erhalten insgesamt

$$\mathbb{E}[S_{PL}] = \mathbb{E}[(n-1)I_{PL}] = (n-1)\prod_{r=1}^{n-1}\sum_{s=0}^{n-r} x^s.$$

$\square$

Bemerkung 5.2 *Für $x = 1$ entspricht die Plackett-Luce-Verteilung der Gleichverteilung auf $\Omega_n = S_n$, denn*

$$\mathbb{P}_1(\sigma) = \prod_{r=1}^{n} \frac{1^{\sigma(r)}}{1^{\sigma(r)} + ... + 1^{\sigma(n)}} = \prod_{r=1}^{n} \frac{1}{n-r+1} = \frac{1}{n!} = \frac{1}{\#S_n},$$

und wir können obiges Ergebnis mit dem aus Satz 2.6 vergleichen:

$$\mathbb{E}[S_{PL}]\Big|_{x=1} = (n-1)\prod_{r=1}^{n-1}\sum_{s=0}^{n-r} 1 = (n-1)\prod_{r=1}^{n-1}(n-r+1) = (n-1)n! = \mathbb{E}[S].$$

Außerdem können wir die zu Beginn von Kapitel 4.2 getätigte Substitution, $x_k = x^k, 1 \leq k \leq n$, rückgängig machen und erhalten allgemeiner:

Korollar 5.3 *In der Situation von Satz 5.1 gilt für reelle, positive $x_1, ..., x_n$*

$$\mathbb{E}[S_{PL}] = \begin{cases} 0 & \text{, falls } \omega \text{ sortiert ist,} \\ (n-1)\prod_{r=1}^{n-1}\sum_{s=0}^{n-r}\frac{x_{r+s}}{x_r} & \text{, sonst.} \end{cases}$$

Beweis : Wir gehen analog zu obigem Beweis vor und erhalten

$$p := \mathbb{P}_x(id) = \prod_{r=1}^{n} \frac{x_r}{x_r + ... + x_n} = \prod_{r=1}^{n-1} \frac{x_r}{x_r + ... + x_n} \in (0,1)$$

als Erfolgswahrscheinlichkeit der somit geometrisch verteilten Zufallsvariable $I_{PL} : S_n \to \mathbb{N}_0$ mit Erwartungswert

$$\mathbb{E}[I_{PL}] = \frac{1}{p} = \prod_{r=1}^{n-1} \frac{x_r + ... + x_n}{x_r},$$

und damit insgesamt

$$\mathbb{E}[S_{PL}] = \mathbb{E}[(n-1)I_{PL}] = (n-1)\prod_{r=1}^{n-1}\sum_{s=r}^{n}\frac{x_s}{x_r} = (n-1)\prod_{r=1}^{n-1}\sum_{s=0}^{n-r}\frac{x_{r+s}}{x_r}.$$

$\square$

5.2 Anzahl der Vergleiche

Zur Bestimmung der Anzahl an Vergleichen, die Bogosort durchführt, um festzustellen, ob eine mittels Plackett-Luce-Verteilung gegebene Permutation sortiert ist, wird zunächst die entsprechende Zufallsvariable $C_{PL} : S_n \to \mathbb{N}$ mit

$$\sigma \mapsto \min \Big\{ \{1 \leq i \leq n-1 | (i,j) \in Inv(\sigma), 2 \leq j \leq n\} \cup \{n-1\} \Big\} \tag{9}$$

eingeführt. Offenbar zählt C_{PL} die in der *Procedure sorted* durchgeführten Vergleiche und

$$\mathbb{E}[C_{PL}] = \sum_{\sigma \in S_n} C_{PL}(\sigma) \prod_{k=1}^{n} \frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{\sigma(n)}}$$

gibt die erwartete Anzahl an benötigten Vergleichen an.

Beispiel 5.4 *Für $n = 3$ berechnen wir*

$$\mathbb{E}[C_{PL}] = \sum_{\sigma \in S_3} C_{PL}(\sigma) \prod_{k=1}^{3} \frac{x^{\sigma(k)}}{x^{\sigma(k)} + ... + x^{\sigma(3)}}$$

$$\begin{aligned}
&= 2\left(\frac{x}{x+x^2+x^3} \cdot \frac{x^2}{x^2+x^3}\right) + 2\left(\frac{x}{x+x^3+x^2} \cdot \frac{x^3}{x^3+x^2}\right) \\
&\quad + 2\left(\frac{x^2}{x^2+x^3+x} \cdot \frac{x^3}{x^3+x}\right) + \left(\frac{x^2}{x^2+x+x^3} \cdot \frac{x}{x+x^3}\right) \\
&\quad + \left(\frac{x^3}{x^3+x^2+x} \cdot \frac{x}{x+x^2}\right) + \left(\frac{x^3}{x^3+x^2+x} \cdot \frac{x^2}{x^2+x}\right)
\end{aligned}$$

$$\begin{aligned}
&= 2\left(\frac{1}{1+x+x^2} \cdot \frac{1}{1+x}\right) + 2\left(\frac{1}{1+x^2+x} \cdot \frac{x}{x+1}\right) \\
&\quad + 2\left(\frac{x}{x+x^2+1} \cdot \frac{x^2}{x^2+1}\right) + \left(\frac{x}{x+1+x^2} \cdot \frac{1}{1+x^2}\right) \\
&\quad + \left(\frac{x^2}{x^2+x+1} \cdot \frac{1}{1+x}\right) + \left(\frac{x^2}{x^2+x+1} \cdot \frac{x}{x+1}\right)
\end{aligned}$$

$$= \frac{(x^3 + x^2 + 2x + 2)(x^2 + 1) + (2x^3 + x)(x + 1)}{(x^2 + x + 1)(x^2 + 1)(x + 1)}$$

$$= \frac{x^4 + 2x^3 + 3x^2 + x + 2}{(x^2 + 1)(x^2 + x + 1)}.$$

Unter Zuhilfenahme von Maple[2] erhalten wir weitere Werte für $n \leq 4$:

$n = 1:$

$$0$$

$n = 2:$

$$1$$

$n = 3:$

$$\frac{x^4 + 2x^3 + 3x^2 + x + 2}{(x^2 + 1)(x^2 + x + 1)}$$

$n = 4:$

$$\frac{x^{16}+4x^{15}+10x^{14}+22x^{13}+37x^{12}+57x^{11}+75x^{10}+90x^9+97x^8+89x^7+82x^6+63x^5+41x^4+30x^3+13x^2+6x+3}{(x^2+1)(x^3+1)(x^2+x+1)(x^3+x^2+1)(x^3+x+1)(x^3+x^2+x+1)}$$

Zunächst betrachten wir wieder das absolute Glied des Zählerpolynoms.

Satz 5.5 *Wird $\mathbb{E}[C_{PL}]$ analog zu (5) als gebrochen rationale Funktion in x dargestellt, also*

$$\mathbb{E}[C_{PL}] = \frac{a_r x^r + ... + a_1 x + a_0}{b_s x^s + ... + b_1 x + b_0},$$

für gewisse $a_0, ..., a_r$ und $b_0, ...b_s$, $r, s \geq 0$, sodass die Darstellung eindeutig ist, so gilt $a_0 = n - 1$.

Beweis : Wir verfahren analog zum Beweis von Satz 4.6 und kürzen zunächst jeden Faktor, um ihn anschließend auf einen gemeinsamen Nenner zu erweitern. Sei dazu wieder

$$N(x) := \prod_{\substack{1 \leq m \leq n-1 \\ 1 \leq k_1 < ... < k_m \leq n-1}} (1 + x^{k_1} + ... + x^{k_m})$$

[2] Quellcode im Anhang

der gemeinsame, von σ unabhängige Nenner und

$$E_\sigma(x) := \prod_{\substack{1\le m\le n-1\\ 1\le k_1<\ldots<k_m\le n-1\\ \nexists k:\{0,k_1,\ldots,k_m\}=\{\sigma(k)-m_k,\ldots,\sigma(n)-m_k\}}} (1+x^{k_1}+\ldots+x^{k_m})$$

das Produkt der Faktoren, mit denen erweitert wird. Dabei ist wie in (6) $m_k = \min\{\sigma(l)|k\le l\le n\}$. Es folgt

$$\begin{aligned}
\mathbb{E}[C_{PL}] &= \sum_{\sigma\in S_n} C_{PL}(\sigma)\prod_{k=1}^{n-1}\frac{x^{\sigma(k)-m_k}}{x^{\sigma(k)-m_k}+\ldots+1+\ldots+x^{\sigma(n)-m_k}}\\
&= \sum_{\sigma\in S_n} C_{PL}(\sigma)\frac{\prod_{k=1}^{n-1}x^{\sigma(k)-m_k}\cdot E_\sigma(x)}{N(x)}\\
&= \sum_{\sigma\in S_n} C_{PL}(\sigma)x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)}\cdot\frac{E_\sigma(x)}{N(x)}.
\end{aligned} \tag{10}$$

Nun ist

$$\sum_{k=1}^{n-1}(\sigma(k)-m_k)=0 \Leftrightarrow \sigma = id,$$

da $m_k\le\sigma(k)$ für alle $1\le k\le n-1$ gilt, also $\sigma(k)=m_k$ für alle k erfüllt sein muss, falls $\sum_{k=1}^{n-1}(\sigma(k)-m_k)=0$ ist, und die umgekehrte Richtung trivialerweise gilt. Damit erhalten wir für das absolute Glied des Zählerpolynoms von $\mathbb{E}[C_{PL}]$

$$a_0 = C_{PL}(id) = n-1.$$

□

Nach weiterer Beobachtung fällt auf, dass der Term $(1+x)$ in der Regel nicht im Nenner von $\mathbb{E}[C_{PL}]$ auftaucht. Die Vermutung liegt also nahe, dass sich dieser Term im Laufe der Berechnung kürzen lässt. Dazu vorab folgendes Lemma.

Lemma 5.6 *Gilt für $\sigma \in S_n$ sowohl $|\sigma(n) - \sigma(n-1)| = 1$ als auch*

$$\sum_{k=1}^{n-1} \Big(\sigma(k) - \min\{\sigma(l) | k \leq l \leq n\}\Big) = 2r \text{ für ein } r \in \mathbb{N}_0,$$

so existiert eine Permutation $\pi \in S_n$ mit

(i) $C_{PL}(\pi) = C_{PL}(\sigma)$,

(ii) $E_\pi(x) = E_\sigma(x)$,

(iii) $|\pi(n) - \pi(n-1)| = 1$ *und*

(iv) $\sum_{k=1}^{n-1} \Big(\pi(k) - \min\{\pi(l) | k \leq l \leq n\}\Big) = 2r \pm 1.$

Beweis : Die obigen Voraussetzungen seien für $\sigma \in S_n$ erfüllt. Wir definieren $\pi \in S_n$ wie folgt

$$\pi(i) := \begin{cases} \sigma(i) & , 1 \leq i \leq n-2, \\ \sigma(n) & , i = n-1, \\ \sigma(n-1) & , i = n. \end{cases}$$

Dann sind (i), (ii) und (iii) offenbar erfüllt. Für (iv) unterscheiden wir zwei Fälle:

1.Fall: Es gilt $\min\{\sigma(n-1), \sigma(n)\} = \sigma(n-1)$. Dann ist

$$\begin{aligned} &\sum_{k=1}^{n-1} \Big(\pi(k) - \min\{\pi(l) | k \leq l \leq n\}\Big) \\ =&\sum_{k=1}^{n-2} \Big(\sigma(k) - \min\{\sigma(l) | k \leq l \leq n\}\Big) + \pi(n-1) - \min\{\pi(n-1), \pi(n)\} \\ =&2r + \sigma(n) - \sigma(n-1) = 2r + 1. \end{aligned}$$

2.Fall: Es gilt $\min\{\sigma(n-1), \sigma(n)\} = \sigma(n)$. Dann ist

$$\begin{aligned} &\sum_{k=1}^{n-1} \Big(\pi(k) - \min\{\pi(l) | k \leq l \leq n\}\Big) \\ =&\sum_{k=1}^{n-2} \Big(\sigma(k) - \min\{\sigma(l) | k \leq l \leq n\}\Big) + \pi(n-1) - \min\{\pi(n-1), \pi(n)\} \\ =&2r - (\sigma(n-1) - \sigma(n)) = 2r - 1. \end{aligned}$$

$\square$

Damit können wir nun unsere Vermutung beweisen:

Satz 5.7 *In der Darstellung*

$$\mathbb{E}[C_{PL}] = \frac{f(x)}{N(x)} \text{ mit } N(x) = \prod_{\substack{1 \leq m \leq n-1 \\ 1 \leq k_1 < \ldots < k_m \leq n-1}} (1 + x^{k_1} + \ldots + x^{k_m})$$

ist $(1+x)$ *ein Teiler von* $f(x)$.

Beweis : Zunächst lässt sich $\mathbb{E}[C_{PL}]$ wie in (10) wie folgt darstellen

$$\mathbb{E}[C_{PL}] = \sum_{\sigma \in S_n} C_{PL}(\sigma) x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)} \cdot \frac{E_\sigma(x)}{N(x)}.$$

Es genügt also zu zeigen, dass $(1+x)$ ein Teiler von

$$\sum_{\substack{\sigma \in S_n \\ (1+x) \nmid E_\sigma(x)}} C_{PL}(\sigma) x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)} E_\sigma(x)$$

ist. Sei also $\sigma \in S_n$ so gewählt, dass $(1+x) \nmid E_\sigma(x)$ gilt. Dann muss $|\sigma(n) - \sigma(n-1)| = 1$ gelten, andernfalls wäre $(1+x)$ ein Faktor von $E_\sigma(x)$. Nach obigem Lemma 5.6 existiert zu jedem Summanden mit geradem Exponenten ein Summand mit ungeradem Exponenten, so dass

$$\sum_{\substack{\sigma \in S_n \\ |\sigma(n)-\sigma(n-1)|=1}} C_{PL}(\sigma) x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)} E_\sigma(x) \Bigg|_{x=-1} = 0$$

gilt, denn, angenommen es gibt $\sigma \neq \sigma' \in S_n$, die die Voraussetzungen des Lemmas erfüllen, also insbesondere

$$\sum_{k=1}^{n-1} \Big(\sigma(k) - \min\{\sigma(l) | k \leq l \leq n\}\Big) = 2r \text{ für ein } r \in \mathbb{N}_0 \text{ und}$$
$$\sum_{k=1}^{n-1} \Big(\sigma'(k) - \min\{\sigma'(l) | k \leq l \leq n\}\Big) = 2r' \text{ für ein } r' \in \mathbb{N}_0,$$

und es gebe zusätzlich ein $\pi \in S_n$ mit

$$\sum_{k=1}^{n-1} \Big(\pi(k) - \min\{\pi(l) | k \leq l \leq n\}\Big) = 2r + 1 = 2r' - 1,$$

so muss auch

$$\begin{aligned}\pi(n) = \sigma(n-1) &= \min\{\sigma(n-1), \sigma(n)\}\\ &= \min\{\pi(n), \pi(n-1)\}\\ &= \min\{\sigma'(n-1), \sigma'(n)\} = \sigma'(n) = \pi(n-1)\end{aligned}$$

gelten, was den Widerspruch und insgesamt die Behauptung liefert. □

Als Zwischenergebnis halten wir fest:

Korollar 5.8 *Mit den bisher eingeführten Notationen und* $\hat{N} := \frac{N(x)}{(1+x)}$ *lässt sich* $\mathbb{E}[C_{PL}]$ *darstellen als*

$$\mathbb{E}[C_{PL}] = \sum_{\sigma \in S_n} C_{PL}(\sigma) x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)} \cdot \frac{E_\sigma(x)}{\hat{N}(x)}. \tag{11}$$

Die Aussage folgt direkt aus obigem Satz. □

In dieser Darstellung können wir den höchsten Exponenten des Zählerpolynoms exakt bestimmen.

Satz 5.9 *Ist* $n \geq 2$ *und wird* $\mathbb{E}[C_{PL}]$ *wie in (11) dargestellt, so ist das Zählerpolynom vom Grad* $(n-2)2^{n-1}$.

Beweis : Zunächst betrachten wir den Exponenten von $x^{\sum_{k=1}^{n-1}(\sigma(k)-m_k)}$. Dieser wird maximal falls $m_k = 1$ für alle $1 \leq k \leq n-1$, also $\sigma(n) = 1$, gilt. In diesem Fall entspricht der Exponent dem Wert

$$\sum_{k=1}^{n-1}(\sigma(k) - m_k) = \sum_{k=1}^{n-1}(\sigma(k) - 1) = \sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}. \tag{12}$$

Widmen wir uns nun dem Grad von $E_\sigma(x)$. Dazu bestimmen wir zunächst den Grad von $\hat{N}$. Betrachten wir dazu die Faktoren mit höchstem Exponenten $1 \leq k \leq n-1$ und insgesamt $2 \leq i \leq n$ Summanden

$$(x^k \underbrace{+ \cdots +}_{i-2} 1),$$

so können wir $i-2$ davon frei wählen und erhalten $\sum_{i=2}^{k+1} \binom{k+1}{i-2} = 2^{k-1}$ Faktoren mit höchstem Exponenten k. Für den maximalen Exponenten in $\hat{N}$ ergibt sich mithilfe der $n-1$-ten Partialsumme der geometrischen Reihe

$$\sum_{k=2}^{n-1} k2^{k-1} = \frac{d}{dt}\left(\sum_{k=2}^{n-1} t^k\right)\Bigg|_{t=2} = \frac{d}{dt}\left(\frac{t^n-1}{t-1} - 1 - t\right)\Bigg|_{t=2}$$

$$= \left(\frac{nt^{n-1}(t-1) - t^n + 1}{(t-1)^2} - 1\right)\Bigg|_{t=2}$$

$$= (n-2)2^{n-1}. \tag{13}$$

Davon müssen wir nun die Werte der jeweils höchsten Exponenten des Nenners von

$$\prod_{k=1}^{n-1} \frac{x^{\sigma(k)-1}}{x^{\sigma(k)-1} + ... + x^{\sigma(n)-1}}$$

abziehen. Um den Grad zu berechnen, wählen wir diese minimal und erhalten hierfür

$$\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}. \tag{14}$$

Mit (12), (13) und (14) folgt insgesamt

$$\frac{n(n-1)}{2} + (n-2)2^{n-1} - \left(\frac{n(n-1)}{2}\right) = (n-2)2^{n-1}.$$

$\square$

Als nächstes bietet es sich an, auch den Leitkoeffizienten des Zählerpolynoms zu betrachten.

Satz 5.10 *Ist $n \geq 2$ und wird $\mathbb{E}[C_{PL}]$ wie in (11) dargestellt, so hat der Leitkoeffizient im Zähler den Wert* 1.

Beweis : Wegen (12) muss $\sigma(n) = 1$ gelten und wegen (14) ist

$$\prod_{k=1}^{n-1}(x^{\sigma(k)-1} + ... + x^{\sigma(n)-1}) = (1+x)(1+x+x^2)\cdots(1+x+...+x^{n-1}).$$

Beides zusammen gilt nur für $\sigma = n(n-1)\cdots 1$ und (9) liefert

$$C_{PL}\big(n(n-1)\cdots 1\big) = 1$$

und damit die Behauptung. □

Zusammenfassend konnten wir mit Korollar 5.8 auch in diesem Abschnitt eine neue Darstellung der erwarteten Anzahl an Vergleichen finden und in dieser Darstellung absolutes Glied, Grad und Leitkoeffizient des Zählerpolynoms exakt bestimmen.

Schlussbetrachtung

In dieser Arbeit wurden mithilfe elementarer Methoden vorrangig randomisierte Sortieralgorithmen hinsichtlich ihrer Effizienz bei beliebig großer, aber stets endlicher Eingabe analysiert.
Beim Überprüfen auf Sortiertheit in *Bogosort* wurden verschiedene Szenarien betrachtet und miteinander verglichen. Auffällig war die Übereinstimmung des Laufzeitverhaltens im Fall einer gegebenen Permutation der Länge n und im Fall einer $n-$elementigen Liste nicht notwendigerweise verschiedener Einträge. Des Weiteren wurde hier die fast sichere Terminierung des Verfahrens festgestellt.
Bei den Analysen von *Quicksort* konnten in der randomisierten und in der deterministischen Variante zwar keine laufzeitbetreffenden, aber dennoch anwendungsorientierte Unterschiede ermittelt werden.
Die Grundvoraussetzung einer gleichverteilt zufällig gewählten Liste im Input wurde bei *Bubble-* und *Bogosort* durch die allgemeinere Annahme einer mittels Plackett-Luce-Verteilung gegebenen Permutation erweitert. Dabei wurden aus wenigen explizit berechneten Anfangswerten allgemein gültige Aussagen entwickelt und bewiesen.
Darüber hinaus wurden Vermutungen erstellt, deren Nachweise sich durchaus als lohnenswert erweisen könnten, schließlich betreffen diese Erkenntnisse nicht nur die hier betrachteten Algorithmen, sondern auch die zugrundeliegenden Strukturen, beispielsweise im Bezug auf die Anzahl der Inversionen in Permutationen.

Anhang

In diesem Abschnitt werden die in den Kapiteln 4 und 5 zur Gewinnung erster Daten genutzten Maple Funktionen aufgelistet. Um die Funktionen zu implementieren muss zunächst das Paket *combinat* mit dem Befehl

```
with(combinat);
```

geladen werden.

Prozedur Plackett-Luce

```
pl := proc(s)
        local n;
        n := nops(s);
        product(x^s[i]/(sum(x^s[j], j = i .. n)), i = 1 .. n);
end proc;
```

Bubblesort

Prozedur Anzahl der Inversionen

```
inv := proc(s)
        local i, j, k, n;
        n := nops(s); j := 0;
        for i to n-1 do
                for k from i+1 to n do
                        if s[k] < s[i] then
                                j := j+1
                        end if;
                end do;
        end do;
        return j;
end proc;
```

Prozedur Erwartungswert

```
erw := proc(n)
        local pn, su, i;
        pn := permute(n); su := 0;
                for i to n! do
                        su := su+inv(pn[i])*pl(pn[i])
                end do;
```

```
        return su;
end proc;
```

Bogosort

Prozedur Anzahl der Vergleiche

```
ver := proc(s)
        local n, i;
        n := nops(s);
                for i to n-1 do
                        if s[i+1] < s[i] then
                                return i
                        end if;
                end do;
        return n-1;
end proc;
```

Prozedur Erwartungswert Vergleiche

```
erwC := proc(n)
        local pn, su, i;
        pn := permute(n); su := 0;
                for i to n! do
                        su := su+ver(pn[i])*pl(pn[i])
                end do;
        return su;
end proc;
```

Literatur

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: *Algorithmen - Eine Einführung.* De Gruyter Oldenbourg, 2013.

[2] Ronald L. Graham, Donald E. Knuth, Oren Patashnik: *Concrete Mathematics: A Foundation for Computer Science.* Addison-Wesley, 2000.

[3] Hermann Gruber, Markus Holzer, Oliver Ruepp: *Sorting the Slow Way: An Analysis of Perversely Awful Randomized Sorting Algorithms.* In: 4th International Conference on Fun with Algorithms, Castiglioncello, Italy, 2007, Lecture Notes in Computer Science 4475, S. 183-197.

[4] Pavel P. Korowkin: *Ungleichungen.* Deutscher Verlag der Wissenschaften, Berlin 1966.

[5] Thomas Schickinger, Angelika Steger: *Diskrete Strukturen: Band 2 Wahrscheinlichkeitstheorie und Statistik.* Springer-Verlag Berlin Heidelberg, 2002.

[6] Angelika Steger: *Diskrete Strukturen: Band 1 Kombinatorik, Graphentheorie, Algebra.* Springer-Verlag Berlin Heidelberg, 2002.

[7] Eli Upfal, Michael Mitzenmacher: *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press New York, 2005.

Eidesstattliche Versicherung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, ganz oder in Teilen noch nicht als Prüfungsleistung vorgelegt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Sämtliche Stellen der Arbeit, die benutzten Werken im Wortlaut oder dem Sinn nach entnommen sind, habe ich durch Quellenangaben kenntlich gemacht. Mir ist bewusst, dass es sich bei Plagiarismus um akademisches Fehlverhalten handelt, das sanktioniert werden kann.

Marburg, den