Randomisierte Algorithmen

Die Verwendung von Zufallszahlen kann manche Algorithmen ganz erheblich beschleunigen! Es ist zwar ein eigener Forschungsgegenstand, ob und wie man geeignete Zufallszahlen überhaupt produzieren kann, aber wir abstrahieren davon und führen einfach eine Operation `WÄHLE` x `ZUFÄLLIG AUS` $\{1,\ldots,n\}$ oder `WÄHLE` x `ZUFÄLLIG AUS` $\{0,1\}$ ein mit der offensichtlichen Semantik.

1. Anwendung: Quicksort

Bei Quicksort haben wir zwei Möglichkeiten der *Randomisierung:*

1) Man kann das Pivot-Element in jedem Schritt zufällig wählen.

2) Man kann die Eingabe zu Beginn „zufällig“ permutieren.

Beiden Techniken ist gemeinsam, dass das Ergebnis jedenfalls immer korrekt ist - unabhängig von den Zufallszahlen!

Primzahltest

Die Überprüfung, ob eine Zahl eine Primzahl ist, ist theoretisch mit dem AKS-Test deterministisch in polynomieller Zeit möglich. Wesentlich effizienter ist aber ein probabilistischer Primzahltest.

Es gibt Tests (z.B. den Fermat-Primzahltest), die für jede große Zahl ausgeben: Ist keine Primzahl oder Ist wahrscheinlich Primzahl. Dabei ist die Aussage im ersten Fall immer korrekt, im zweiten Fall ist ein Irrtum mit einer festgelegten Wahrscheinlichkeit erlaubt.

Selbst wenn ein solcher Test eine Fehlerwahrscheinlichkeit von 25% hat, ist er durchaus brauchbar. Man kann leicht nachrechnen, dass bereits mit 5 Wiederholungen die Fehlerquote auf 0,1% gesenkt werden kann; wie sieht es mit 100 Wiederholungen aus?

Bei 100 Wiederholungen ist die Fehlerwahrscheinlichkeit so gering, dass man die Möglichkeit eines Fehlers praktisch ohne Risiko ausschließen kann.

Verstärkung der Erfolgswahrscheinlichkeit

Die eben mehr oder weniger intuitiv durchgeführte Verbesserung eines probabilistischen Algorithmus durch Wiederholung kann man formal so beschreiben:

L sei die zu erkennende Sprache, und es liege ein probabilistischer Algorithmus für L vor, der folgendes Verhalten zeigt:

Input $x \in L$:	Der Algorithmus gibt JA aus.
Input $x \notin L$:	Der Algorithmus gibt NEIN mit W'keit $\geq \frac{1}{2}$ aus.

Beachte: Das bedeutet, dass eine NEIN-Antwort des Algorithmus immer zuverlässig ist, während die JA-Antworten nicht sicher sind.

Nun wiederholen wir den Algorithmus n mal mit immer wieder neu gewählten Zufallszahlen. Wir schließen $x \in L$, falls n mal Antwort JA gegeben wurde, $x \notin L$, sonst.

Auch hier ist eine NEIN-Antwort immer korrekt, während eine JA-Antwort mit Wahrscheinlichkeit $\leq \frac{1}{2^n}$ falsch sein kann.

Verstärkung bei zweiseitigem Fehler

Die Verbesserung durch Wiederholung scheint auf den ersten Blick nur wegen der Einseitigkeit des Fehlers zu funktionieren.

Tatsächlich kann man aber auch bei zweiseitigem Fehler einen ähnlichen Effekt durch Wiederholung erzielen. Die Rechnung wird hier allerdings ein bisschen komplizierter.

Beispiel: Der ursprüngliche Algorithmus produziere das richtige Ergebnis mit Wahrscheinlichkeit $\frac{2}{3}$. Wir wiederholen dreimal und nehmen als Ergebnis die Mehrheitsentscheidung. Also geben wir das richtige Ergebnis mit Wahrscheinlichkeit $(\frac{2}{3})^3 + 3 \cdot (\frac{2}{3})^2 \cdot \frac{1}{3} = \frac{20}{27}$ aus.

Bitte überprüfen Sie, ob Sie das Zustandekommen dieser Rechnung verstanden haben.

Allgemein kann man zeigen, dass die t-fache Wiederholung eines probabilistischen Algorithmus, der mit Wahrscheinlichkeit $> (1 - \delta)$ das korrekte Ergebnis liefert, maximale Fehlerwahrscheinlichkeit δ' hat, wobei $\delta' = \left[2 \cdot \sqrt{\delta(1-\delta)}\right]^t$ ist (für $0 < \delta < \frac{1}{2}$).

Monte Carlo Algorithmen

Bei probabilistischen Algorithmen werden oft zwei Konzepte unterschieden. Das erste dieser Konzepte stellen die

Monte Carlo Algorithmen

dar.

Das charakteristische Merkmal dieser Sorte probabilistischer Algorithmen ist die Tatsache, dass sie ein falsches Ergebnis liefern dürfen. Allerdings sollte die Wahrscheinlichkeit, das korrekte Ergebnis zu liefern, möglichst hoch sein.

Die Wahrscheinlichkeit ist hierbei immer zu verstehen bzgl. der Gleichverteilung bei dem zugrunde gelegten Zufallsexperiment.

Als ein Beispiel für diese Art der probabilistischen Algorithmen haben wir gerade die Primzahltests behandelt.

Las Vegas Algorithmen

sind die andere Sorte probabilistischer Algorithmen. Hier sind keine Fehler erlaubt. Allerdings besteht bei solchen Algorithmen im Normalfall die Option, keine Ausgabe zu machen.

Was ist bei diesen Algorithmen dann noch zufällig?

Zum einen kann die Laufzeit eines Las Vegas Algorithmus von den verwendeten Zufallswerten abhängen. Darüberhinaus kann es sein, dass der Algorithmus keine Ausgabe produziert und folglich (mit neuen Zufallswerten) wiederholt werden muss.

Welches Ziel verfolgt man eigentlich in diesen Fällen durch die Verwendung des Zufalls?

Antwort: Die erwartete Rechenzeit soll minimiert werden!

2. Teil: Diskrete Strukturen

Die nächsten 23 Einheiten betreffen Inhalte aus dem Bereich der diskreten Strukturen – mit Betonung auf Resultaten mit starkem Bezug zur Informatik:

- Einheiten 18-22: Algebraische und zahlentheoretische Algorithmen
- Einheiten 23-25: Graphen
- Einheiten 26-30: RSA, Euler, Fibonacci
- Einheiten 31-33: Wachstumsabschätzungen
- Einheiten 34-35: Diskrete Wahrscheinlichkeit
- Einheiten 36-38: Kombinatorik
- Einheiten 39-41: Satz von Ramsey, Zusammenfassung

Zahlensysteme

Kenntnis der Zahlenbereiche $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ setzen wir voraus.

Axiomatische Einführung von $\mathbb{N}$ über Peano-Axiome.

$\mathbb{Z}$ aus $\mathbb{N}$ leicht abzuleiten.

Wie wird $\mathbb{Q}$ definiert?

$\mathbb{R}$ ist der erste *überabzählbare* Zahlenbereich.

Von $\mathbb{R}$ gelangt man zu $\mathbb{C}$ durch Hinzunahme von $i = \sqrt{-1}$.

In $\mathbb{N}$ gibt es die wichtige Teilmenge der *Primzahlen*.

Halbgruppen und Monoide

Sei M eine Menge und $\circ$ eine zweistellige Verknüpfung, d.h.

$$\circ : M \times M \to M, \quad (m, n) \mapsto \circ(m, n)$$

Man schreibt dann oft $m \circ n$ statt $\circ(m, n)$.

Wenn $\circ$ eine assoziative Verknüpfung ist, sagen wir:

$(M, \circ)$ bildet eine Halbgruppe.

Hierbei bedeutet *assoziativ*, dass $\forall a, b, c \in M : (a \circ b) \circ c = a \circ (b \circ c)$.

Wenn in einer Halbgruppe $(M, \circ)$ außerdem noch ein neutrales Element existiert, d.h. ein $e \in M$, für das $a \circ e = e \circ a = a$ für alle $a \in M$, dann sagen wir

$(M, \circ)$ bildet ein Monoid.

Manchmal schreibt man auch $(M, \circ, e)$, um das neutrale Element herauszuheben.

Gruppen

Wenn ein Monoid für jedes Element ein sogenanntes *beidseitiges Inverses* enthält, so nennen wir es eine Gruppe:

$$(G, \circ, e)$$

Wir zählen noch einmal alle Bedingungen an die Gruppenstruktur auf, man nennt sie auch *Gruppenaxiome*:

$\circ : G \times G \to G, \quad (g_1, g_2) \mapsto g_1 \circ g_2$

$(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$ für alle $g_1, g_2, g_3 \in G$

$g \circ e = e \circ g = g$ für alle $g \in G$

Für jedes $g \in G$ gibt es ein $h \in G$ mit $g \circ h = h \circ g = e$

Gilt außerdem $g \circ h = h \circ g$ für alle $g, h \in G$, dann ist G eine kommutative oder abelsche Gruppe.

Ringe und Körper

Ein Ring R besteht aus einer *additiven* Gruppe mit neutralem Element 0, d.h. $(R, +, 0)$ ist eine abelsche Gruppe, sowie einer zusätzlichen Operation $\cdot$ (der Multiplikation), so dass $(R, \cdot, 1)$ mit einem Element $1 \in R$ ein Monoid bildet und die Distributivgesetze gelten:

$$a \cdot (b + c) = a \cdot b + a \cdot c \quad \text{und} \quad (a + b) \cdot c = a \cdot c + b \cdot c.$$

Ein Körper K ist ein *kommutativer Ring* (also ein Ring, bei dem auch die Multiplikation kommutativ ist, d.h. $a \cdot b = b \cdot a$ für alle a, b), bei dem die Menge $K \setminus \{0\}$ mit der Multiplikation ebenfalls eine Gruppe bildet, d.h. für jedes $a \in K \setminus \{0\}$ gibt es ein Inverses $b \in K \setminus \{0\}$, so dass $a \cdot b = b \cdot a = 1$ gilt.

Unterstrukturen

Wenn X eine algebraische Struktur ist, dann ist eine Teilmenge $Y \subseteq X$ eine Unterstruktur, falls Y ebenfalls die von X geforderten Struktureigenschaften hat.

So gibt es also Untergruppen in Gruppen, Untermonoide in Monoiden, Unterhalbgruppen in Halbgruppen, und natürlich auch Unterringe in Ringen und Unterkörper in Körpern.

Beispiele:

In der Gruppe $(\mathbb{Z}, +, 0)$ ist für festes $n \in \mathbb{N}$ die Menge $n\mathbb{Z} = \{n \cdot z \mid z \in \mathbb{Z}\}$ eine Untergruppe.

Im Körper $\mathbb{R}$ ist $\mathbb{Q}$ ein Unterkörper.

Im Monoid aller Wörter über dem Alphabet $\{a, b, c\}$ bilden die Wörter, die nur aus a's und b's bestehen, ein Untermonoid.

Homomorphismen

Ein Homomorphismus ist eine *strukturerhaltende Abbildung.*

So ist z.B. ein Monoid-Homomorphismus eine Abbildung φ von einem Monoid $(M_1, \circ_1, e_1)$ in ein Monoid $(M_2, \circ_2, e_2)$ mit der Eigenschaft, dass $\varphi(m \circ_1 m') = \varphi(m) \circ_2 \varphi(m')$ und $\varphi(e_1) = e_2$ gilt.

In ähnlicher Weise gibt es auch Gruppenhomomorphismen, Ringhomomorphismen und Körperhomomorphismen.

Homomorphismen mit zusätzlichen Eigenschaften haben spezielle Namen: Monomorphismus (injektiver Homomorphismus), Epimorphismus (surjektiver Homomorphismus), Isomorphismus (bijektiver Homomorphismus).