Algorithmen und Berechenbarkeit 01

Mitschrieb

17.10.2017

Abstract

Organisatorisches, Randomisierte Algorithmen (Closest Pair). **Das hier ist kein Mitschreibservice.** Bitte überprüft insbesondere die Formeln auf ihre Richtigkeit und korrigiert falsche Angaben. Auch an der Optik darf geschraubt werden.

Organisatorisches

Es gibt einen Schein. Um diesen Schein zu bekommen, müssen die folgenden Voraussetzungen erfüllt sein:

- 50% der Punkte aus den Übungen
- Bestehen von 2 MC-Tests

Pro Übungsblatt wird in der Regel eine Woche Bearbeitungszeit eingeräumt. Wer Aufgaben abgibt, "votiert" automatsich für diese Aufgaben. Falls die Aufgaben im Zweifel unzureichend vorgerechnet werden, gibt es 0 Punkte für das gesamte Blatt.

Randomisierte Algorithmen

Randomisierte Algorithmen sind Algorithmen, welche unter Nutzung einer Zufallsquelle (z.B. Münzwurf, Zufallsgenerator) Probleme lösen. In manchen Fällen sind diese Algorithmen einfacher und effizienter als die entsprechenden deterministischen Algorithmen.

Randomisierte Algorithmen werden nach "dem Verbrauch von Zufall" bewertet, denn Zufall zu erzeugen, ist nicht so ohne Weiteres möglich (Zufall zu "erzeugen" ist *teuer*). Im Gegensatz dazu bewertet man "normale" Algorithmen nach Platz- und Zeitbedarf.

Closest Pair (Randomisiert)

Gegeben seien n Punkte im R^2

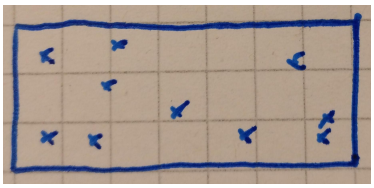

Figure 1: Skizze Closest Pair Ausgangslage

Ziel: Finde das Paar mit dem kleinsten Abstand zueinander.

Naiver Ansatz

- Betrachte P_1 und berechne Distanzen zu $P_2, P_3, ..., P_n$
- Betrachte P_2 und berechne Distanzen zu $P_3, P_4, ..., P_n$
- ...

Das würde eine Laufzeit von

$$O\left(\sum_{i=1}^{n-1}\right) = O\left(\frac{(n-1)\cdot n}{2}\right) = O(n^2)$$

ergeben. Ein Prozessor mit 1 GHz Taktfrequenz schafft 1.000.000 Instruktionen pro Sekunde. Das führt zu folgender Tabelle:

n	Rechnung		Ergbnis
$= 100$	$100^2 \cdot 10^{-9}$	$= 10^{-5}$	10 μs
$= 1000$	$1000^2 \cdot 10^{-9}$	$= 10^{-3}$	1 ms
$= 10000$	$10000^2 \cdot 10^{-9}$	$= 10$	10 s
$= 1000000$	$1000000^2 \cdot 10^{-9}$	$= 10^3$	15 min

$O(n^2)$ ist für das Problem nicht praktikabel.

Randomisierter Algorithmus für CP

Ein randomisierter Algorithmus mit *erwarteter Laufzeit* (Varianten: Glück & Pech) hat eine Laufzeit von $O(n)$. Er wird aus folgendem inkrementellen Ansatz hergeleitet:

Zuerst wird die Datenmenge $P = \{P_1, P_2, P_3, ..., P_n\}$ betrachtet. Sei nun δ_i die CP-Distanz in der Menge P. Es kommt nun ein weiterer Punkt P_{i+1} hinzu.

Eine einfache Möglichkeit, den neuen geringsten Abstand zu finden, wäre, wenn man die Abstände vom neuen Punkt zu allen anderen Punkten vergleicht und überprüft, ob es einen kleineren Abstand als das bisherige δ_i gibt.

Besser wäre jedoch Folgendes: Angenommen, δ_i ist bestimmt. So kann nun ein Gitter erzeugt werden, das eine Maschenweite (Breite der Zeilen/Spalten) von genau δ_i hat.

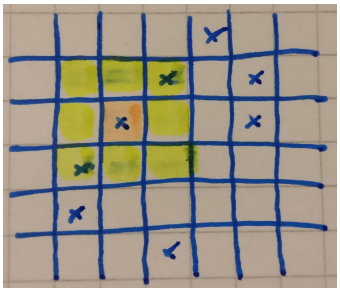

Figure 2: Einfügen eines neuen Elements in das Gitter

Das Einfügen folgt einem einfachen Ablauf

1. Lokalisiere P_{i+1} im Gitter (orange)
2. Inspiziere alle Punkte im Feld von P_{i+1} sowie alle Felder um P_{i+1} herum (gelb)
3. - $\delta_{i+1} = \delta_i \rightarrow$ Nichts mehr tun $\rightarrow O(1)$
   - $\delta_{i+1} < \delta_i \rightarrow$ Baue Gitter mit neuer Maschenweite $\delta_{i+1} \rightarrow O(n^2)$ im schlimmsten Fall

Da es auch Fälle geben kann, in denen P_{i+1} immer einen kleineres δ hat als davor (Liste nach größtem Abstand geordnet), empfiehlt es sich, die Punkte immer in zufälliger Reihenfolge "einzufügen". So bleibt eine Wahrscheinlichkeit von $\frac{2}{i+1}$ dafür, dass der nächste Punkt ein CP-Punkt ist.

Die erwarteten Kosten des Einfügens sind

$$\leq \underbrace{\frac{2}{i+1} \cdot O(n)}_{schlechter Fall} + \underbrace{O(1)}_{guter Fall} = O(1) + O(1) = O(1)$$

Damit lässt sich der Erwartungswert berechnen zu

$$E\left[\sum_{i=1}^{n}(Kosten\ für\ Einfügen\ von\ P_i)\right] = \sum_{i=1}^{n} E[Kosten\ für\ Einfügen\ von\ P_i] = \sum_{i=1}^{n} O(1) = O(n)$$

Lemma: Die Wahrscheinlichkeit, beim Einfügen von P_{i+1} das Gitter neu aufbauen zu müssen, ist $< \frac{2}{i+1}$.

Beweis: Das Gitter muss genau dann neu aufgebaut werden, wenn P_{i+1} einer der beiden Punkte ist, welche das CP in der Menge der ersten $i+1$ Punkte bestimmen. Jeder der ersten $i+1$ Punkte ist mit gleicher Wahrscheinlichkeit der P_{i+1}.

Falls CP eindeutig:

$$\rightarrow P(Gitter\ muss\ neu\ aufgebaut\ werden) = \frac{2}{i}$$

Closest Pair mit deterministischem Algorithmus

Closest Pair kann in $O(n \cdot log(n))$ berechnet werden. Es kann sogar gezeigt werden, dass CP mindestens $\Omega(n \cdot log(n))$ braucht.

n	Rechnung	Ergbnis
$= 1000000$	$10^6 \cdot 6 \cdot 10^{-9}$	6 ms

Anhang

Zufallsvariable und Erwartungswert

Sei Y eine Zufallsvariable, zum Beispiel *Augenzahl beim Wurf mit einem normalen und fairen Würfel.* Der Erwartungswert berechnet sich zu

$$E[X] = \sum Ereignis \cdot P(Ereignis)$$

Für Y also: $\frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = 3,5$