2. Aufgabenblatt zur Vorlesung

Randomisierte Algorithmen WiSe 2016/17

Wolfgang Mulzer, Paul Seiferth, Max Willert

Abgabe bis zum 02. November 2016, 12 Uhr, in den Briefkasten neben Raum 111

Aufgabe 1 Details zur Implementierung des engste Paares *10 Punkte*

(a) Gegeben sei ein Feld `A` mit n paarweise verschiedenen Elementen. Beschreiben Sie einen Algorithmus, der eine zufällig gleichverteilte Permutation von `A` liefert. Jede Permutation soll mit Wahrscheinlichkeit $1/n!$ gewählt werden.

Ihr Algorithmus soll die Funktion `random(n)` benutzen, welche eine zufällig gleichverteilte Zahl aus der Menge $\{1, \dots, n\}$ liefert. Die Laufzeit soll $O(n)$ betragen.

(b) Ein Gitterwörterbuch ist ein abstrakter Datentyp, der zwei Operationen unterstützt: (i) `insert(i, p)` fügt den Punkt $p \in \mathbb{R}^2$ in das i-te Standardgitter ein und (ii) `lookup(i, (a,b))` liefert die Menge aller Punkte, die in der Zelle (a, b) des i-ten Standardgitters gespeichert sind. Dabei werden die Punktmengen in den verschiedenen Standardgittern unabhängig voneinander verwaltet.

Beschreiben Sie eine Implementierung des Gitterwörterbuchs, die $O(\log n)$ Zeit pro Operation benötigt.

Aufgabe 2 Ein alternativer Algorithmus für das engste Paar *15 Punkte*

Sei P eine Menge von n Punkten in der Ebene, so dass alle $\binom{n}{2}$ Abstände paarweise verschieden sind und so dass alle Koordinaten positiv sind. Sei $k = \lfloor \log n \rfloor - 1$. Wir berechnen zufällig Punktmengen $P = P_0 \supset P_1 \supset P_2 \supset \cdots \supset P_k$, indem wir für $i = 1, \dots, k$ die Menge P_i als zufällige Teilmenge von P_{i-1} mit genau $\lfloor n/2^i \rfloor$ Elementen wählen.

Der Algorithums legt für $i = k, \dots, 0$ ein Gitterwörterbuch G_i für P_i an, so dass jede Zelle in G_i höchstens einen Punkt enthält und so dass die Zellen für das engste Paar Abstand $O(1)$ besitzen. Der Algorithmus geht in Runden vor. Die Runden sind nummeriert von k bis 1. In Runde i wird das Gitterwörterbuch G_{i-1} berechnet, unter der Annahme, dass das Gitterwörterbuch G_i verfügbar ist. Das geht so: (i) füge alle Punkte aus P_{i-1} in G_i ein; (ii) bestimme mit einem brute-force Verfahren für die Punkte in jeder Gitterzelle eine optimale neue Gitterbreite; (iii) berechne daraus die Gitterbreite für G_{i-1}; und (iv) füge alle Punkte aus P_{i-1} in G_{i-1} ein.

(a) Beschreiben Sie den Algorithmus in allen Details. Gehen Sie insbesondere darauf ein, wie man am Schluss das engste Paar findet. Die Anzahl der Operationen auf dem Gitterwörterbuch soll $O(n)$ im worst-case betragen. Die Zeit pro Gitterzelle kann quadratisch in der Anzahl der enthaltenen Punkte sein.

(b) Sei Z_i die Anzahl der Schritte, die der Algorithmus in Runde i benötigt. Zeigen Sie: Falls $\mathbf{E}[Z_i] = O(|P_{i-1}|)$ ist, dann benötigt der Algorithmus im Erwartungswert $O(n)$ Operationen.

(c) Sei $Q \subseteq P$ mit $|Q| = \lfloor n/2^{i-1} \rfloor$ fixiert. Zeigen Sie:

$$\mathbf{E}[Z_i \mid P_{i-1} = Q] \leq \sum_{p \in Q} \mathbf{E}[X_p \mid P_{i-1} = Q],$$

wobei X_p die Anzahl der Punkte aus P_{i-1} ist, die in der Zelle von G_i liegen, die p enthält.

(d) Zeigen Sie $\mathbf{E}[X_p \mid P_{i-1} = Q] = O(1)$.

Hinweis: Schätzen Sie zunächst die Wahrscheinlichkeit $\mathbf{Pr}[X_p \geq r \mid P_{i-1} = Q]$ ab. Wenn $X_p \geq r$ ist, dann muss die Zelle in G_i, welche p enthält, mindestens r Punkte aus P_{i-1} enthalten. Verstehen sie genau, wie das passieren kann, und schätzen Sie die Wahrscheinlichkeit für dieses Ereignis ab.

(e) Folgern Sie $\mathbf{E}[Z_i] = O(|P_{i-1}|)$ und schließen Sie die Laufzeitanalyse ab.

Aufgabe 3 MinCuts *5 Punkte*

(a) Benutzen Sie die Analyse des Algorithmus MinCut1, um zu zeigen, dass ein Graph mit n Knoten höchstens $\binom{n}{2}$ minimale Schnitte besitzt.

(b) Geben sie für jedes $n \geq 3$ einen Graphen an, der $\binom{n}{2}$ minimale Schnitte besitzt.