Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

(Kurz-)Übungsblatt 7

Vorlesung Algorithmentechnik im WS 08/09

Ausgabe 04. Februar 2008
Abgabe 12. Februar, 15:30 Uhr (im Kasten vor Zimmer 319, Informatik-Hauptgebäude, 3. OG)

Bitte schreiben Sie nur Ihren Namen und keine Matrikelnummer auf Ihr Übungsblatt. Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Problem 1: Bounding Box

Die *bounding box* einer Menge von Punkten in der Ebene ist das kleinste achsenparallele Rechteck, das alle Punkte der Menge beinhaltet. Wir betrachten folgenden Algorithmus:

Algorithmus 1 : RandomizedBoundingBox(S)

Eingabe : Menge S von Punkten in $\mathbb{R} \times \mathbb{R}$
Ausgabe : Bounding Box B von S

Berechne eine zufällige Permutation $p_1, \ldots p_n$ der Punkte in S
B $\leftarrow$ bounding box von $\{p_1, \ldots, p_4\}$
if *alle* p_i *sind in* B **then**
  return B
else
  return RandomizedBoundingBox(S)

Wir nehmen vereinfachend an, dass genau 4 Punkte auf dem Rand der bounding box von S liegen. Berechne die erwartete Laufzeit des Algorithmus. Schildere dazu eine einfache Vorgehensweise, die überprüft, ob ein Punkt in der bounding box liegt.

Problem 2: Zufälliges Element - nicht gleichverteilt

Sei $S = \{x_1, \ldots, x_n\}$ eine Menge von Elementen mit Gewichtsfunktion weight : $S \to \{1, \ldots n\}$. Es sei $W := \sum_{i=1}^{n} \text{weight}(x_i)$. Beschreibe einen Algorithmus, der ein zufälliges Element x_i aus S mit Wahrscheinlichkeit $\text{weight}(x_i)/W$ wählt. Die Laufzeit des Algorithmus soll in $O(n)$ liegen. Dazu darf eine Routine $\text{random}(a, b)$ benutzt werden, die in Zeit $O(1)$ eine ganze Zahl gleichverteilt aus der Menge $\{a, a+1, \ldots, b\}$ wählt.

Problem 3: Fingerprints

Gegeben ist folgender Monte-Carlo Algorithmus

Algorithmus 2 : Überprüfe ob Bitfolgen gleich sind

Eingabe : Bitfolgen $a_1 \dots a_n$ und $b_1 \dots b_n$
Ausgabe : Entscheidung ob Bitfolgen gleich sind
$p \leftarrow$ <Primzahl, zufällig gleichverteilt aus denen kleiner oder gleich $n^2 \log n^2$ gewählt>
$a \leftarrow \sum_{i=1}^{n} a_i 2^{i-1}$
$b \leftarrow \sum_{i=1}^{n} b_i 2^{i-1}$
if $a \mod p \equiv b \mod p$ **then**
  ∟ **return** JA
else
  ∟ **return** NEIN

Zeigen Sie, dass die Wahrscheinlichkeit, dass der Algorithmus ein falsches Ergebnis liefert in $O(1/n)$ liegt. Benutzen Sie dafür die folgenden beiden Resultate:

Satz 3.1 (Chebyshew)**.** *Für die Anzahl $\pi(n)$ der Primzahlen kleiner oder gleich n gilt*

$$\frac{7}{8}\frac{n}{\ln n} \leq \pi(n) \leq \frac{9}{8}\frac{n}{\ln n}$$

Es ist also $\pi(n) \in \Theta(n/\ln n)$.

Satz 3.2. *Die Anzahl k verschiedener Primteiler einer Zahl kleiner oder gleich 2^n ist höchstens n.*