
Algorithmen für die Speicherhierarchie

Abgabetermin: 1.7.2009 vor der Vorlesung

Aufgabe 1

Gegeben sei ein Array von n Pointern, die eine Liste repräsentieren. In der Vorlesung wurde vorgestellt, wie ein effizientes List-Ranking realisiert werden kann. Nun enthalte die Liste zusätzlich noch Werte von denen die Prefix-Summen ermittelt werden sollen. Welche obere Schranke können Sie dafür zeigen?

Aufgabe 2

Zur Berechnung des List-Rankings wurde jeweils ein Independent Set erzeugt. Dazu wurde jedem Listenelement l mit Rang r ein Zufallsbit $x_r \in \{0, 1\}$ gleichverteilt zugewiesen. Falls dann gilt $x_r = 1$ und $x_{r-1} = 0$ wird l zum Independent Set hinzugefügt.

- Bestimmen Sie die erwartete Größe eines so erzeugten Independent Set aus einer Liste mit n Elementen.
- Im Algorithmus zum List-Ranking werden rekursiv die Elemente des Independent Set aus der Liste entfernt und auf der resultierenden Liste erneut ein Independent Set bestimmt. Wieviele Zufallsbits müssen im Erwartungswert erzeugt werden bis alle Elemente aus der Liste entfernt wurden?
- Schätzen Sie mit der Chernoff-Ungleichung nach oben die Wahrscheinlichkeit ab, dass mehr als $5n$ Zufallsbits benötigt werden bis alle Elemente entfernt wurden.

Hinweis für b) und c): Beachten Sie, dass die Zufallsvariablen, ob ein Element dem Independent Set hinzugefügt wird nicht unabhängig sind. Zur Abhilfe können Zufallsvariablen $y_r = x_r \text{ XOR } x_{r-1}$ mit $x_0 = 0$ definiert werden, von denen mindestens die Hälfte der Elemente mit Wert $y_r = 1$ dem Independent Set angehören.

Aufgabe 3

Ein Array aus Pointern muss nicht zwangsläufig eine Liste darstellen.

- Wie kann man überprüfen, ob es sich tatsächlich um eine Liste handelt?
- Welche anderen Strukturen wären ansonsten möglich? Wie können Anzahl und Größe der Teilstücke ggf. bestimmt werden?
- Definieren Sie sich auch für solche Strukturen ein Ranking und versuchen Sie dieses möglichst effizient zu bestimmen.

Betrachten Sie natürlich auch die Komplexität der Aufgaben.