2. Aufgabenblatt zur Vorlesung

Höhere Algorithmik — WiSe 2016/17

Wolfgang Mulzer, Boris Klemz

Abgabe bis zum 04. November 2016, 12 Uhr, in den Briefkasten neben Raum 111

Aufgabe 1 Gewichteter Median *10 Punkte*

Sei $S = \{s_1, s_2, \dots, s_n\}$ eine Menge von n paarweise verschiedenen Elementen aus einem total geordneten Universum. Seien w_1, w_2, …, w_n positive Gewichte, so dass das Element s_i Gewicht w_i hat. Es gelte $\sum_{i=1}^n w_i = 1$. Der *gewichtete Median* von S ist das Element $s_k \in S$, so dass gilt:

$$\sum_{i:s_i<s_k} w_i \leq 1/2$$

und

$$\sum_{i:s_i>s_k} w_i < 1/2.$$

(a) Angenommen, Sie haben eine Funktion, welche den gewichteten Median in Zeit $T(n)$ bestimmt. Zeigen Sie, wie man den (normalen) Median in Zeit $O(n) + T(n)$ berechnen kann.

Hinweis: Finden Sie eine geschickte Wahl der Gewichte.

(b) Zeigen Sie, wie man den gewichteten Median in $O(n \log n)$ Zeit berechnen kann.

Hinweis: Verwenden Sie einen geeigneten Sortieralgorithmus.

(c) Zeigen Sie, wie man den gewichteten Median in $O(n)$ Zeit finden kann, wenn ein ein Linearzeitalgorithmus zum Finden des normalen Medians zur Verfügung steht.

Hinweis: Passen Sie den generischen SELECT-Algorithmus geeignet an.

Aufgabe 2 Analyse des BFPRT-Algorithmus *10 Punkte*

In der Vorlesung wurde der BFPRT-Algorithmus vorgestellt, der in linearer Zeit das Auswahlproblem löst. Dazu wird die Eingabe in 5-er Blöcke aufgeteilt. Bleibt die Laufzeit linear, wenn man die Eingabe in 7-er Blöcke unterteilt? Welche Laufzeit erhält man, wenn man 3-er Blöcke verwendet (in der Θ-Notation)? Begründen Sie jeweils Ihre Antwort.

(*freiwillig, 5 Zusatzpunkte*) Man könnte die Eingabe auch für festes ungerades k in k-er Blöcke aufteilen. Wie verhält sich die Laufzeit in Abhängigkeit von k?

Bitte wenden

Aufgabe 3 Schmutzige Tricks mit dem Einheitskostenmaß *10 Punkte*

Wenn die Zahlen in den Registern der RAM zu groß werden, wird das Einheitskostenmaß unrealistisch. Wir wollen nun ein Beispiel sehen, wie man dadurch in Algorithmen schummeln kann.

(a) Für zwei ganzzahlige Vektoren $x = (x_1; x_2; \ldots; x_n)$ und $y = (y_1; y_2; \ldots; y_n)$ mit $0 \leq x_i, y_i \leq M$ und einen Wert $u > M$ betrachten wir die Zahlen

$$a = x_1 u^n + x_2 u^{n-1} + \cdots + x_n u^1$$

und

$$b = y_1 u^n + y_2 u^{n-1} + \cdots + y_n u^1.$$

Zeigen Sie: $a = b$ genau dann, wenn $x = y$ ist; $a < b$ genau dann, wenn x lexikographisch kleiner als y ist.

(Man kann also den Vergleich beliebig langer Vektoren durch den Vergleich einzelner Zahlen ersetzen.)

(b) Anwendung: Entwerfen Sie einen Algorithmus, der für zwei gegebene Folgen nichtnegativer ganzer Zahlen $x = (x_1; x_2; \ldots; x_m)$ und $y = (y_1; y_2; \ldots; y_n)$ in "linearer" Zeit entscheidet, ob x als Teilfolge $(y_{i+1}; y_{i+2}; \ldots; y_{i+m})$ in y vorkommt $(0 \leq i \leq n - m)$. Was ist die Laufzeit Ihres Algorithmus im EKM und im LKM?

Bemerkung: Mit Hilfe von Hashing kann man aus dieser Idee einen "seriösen" Algorithmus zum Finden von Teilfolgen entwickeln. Dies ist der Algorithmus von Karp-Rabin.