Datenstrukturen und effiziente Algorithmen
Blatt 4

Markus Vieth, David Klopp, Christian Stricker

18. November 2015

Nr.1

Psoidocode:

Eingabe: Integer k für Anzahl an Farben, Integer l für Anzahl an Muster. Ein 2x2n Array. In der ersten Zeile stehen die Farben in Integer codiert, in der zweiten Zeile analog für die Muster(bei 5 verschiedenen Muster/Farben gibt es die Integer 0-4).
Man iterriert über das Eingabearray und speichert die Anzahl der jeweiligen Farben in ein Array. Dann wird nochmal über das Array iterriert, wenn an der i-ten Stelle, die Socke mit falsche Farbe n steht, wird diese mit einer anderen Socke im n Bereich vertauscht.
Danach wird für jeden Farbbereich analog zum Farbensortieren nach Muster sortiert(Anzahl der Muster bestimmen, an die richtige Stelle tauschen).
Damit kommt eine Laufzeit von O(n+l+k)= O(n) zustande.
O(4*n), da man 2 mal über das ganze Array iteririert, um die Anzahl (an Farben/Muster) zu bestimmen
O(8*n), da höchstens n-mal vertauscht wird, 2n Zugriffe zum Zwischenspeichern, 4n Zugriffe zum Überschreiben und 2n Zugriffe für die Zwischenspeicherungen wieder zu speichern.
Analog beim Mustertauschen.

Nr.2

a)

Überlegung analog zur Vorlesung:

m=3

$$\exists\ \frac{2n}{6} \text{ Elemente } \leq p \Rightarrow \text{Es existieren maximal } \frac{4n}{6} \text{ Elemente } \geq p$$

$$\exists\ \frac{2n}{6} \text{ Elemente } \geq p \Rightarrow \text{Es existieren maximal } \frac{4n}{6} \text{ Elemente } \leq p$$

m=7

$$\exists\ \frac{4n}{14} \text{ Elemente } \leq p \Rightarrow \text{Es existieren maximal } \frac{10n}{14} \text{ Elemente } \geq p$$

$$\exists\ \frac{4n}{14} \text{ Elemente } \geq p \Rightarrow \text{Es existieren maximal } \frac{10n}{14} \text{ Elemente } \leq p$$

b)

Überlegung analog zur Vorlesung:

m=3

$$T(n) = T\left(\frac{n}{3}\right) + n + T\left(\frac{2n}{3}\right)$$

Akra-Bazzi:

$$g(n) = n,\ a_1 = a_2 = 1,\ b_1 = 3\ b_2 = \frac{3}{2}$$

$$1 = \left(\frac{1}{3}\right)^\alpha + \left(\frac{2}{3}\right)^\alpha$$

$$\Leftrightarrow \alpha = 1$$

$$T(n) = n\left(1 + \int_1^n \frac{x}{x^2}dx\right) = n\left(1 + \int_1^n \frac{1}{x}dx\right) = n\left(1 + [\ln(x)]_1^n\right) = n + n\ln(n) \in O(n\log(n)) \neq O(n)$$

Somit ist die Laufzeit für $m = 3$ nicht linear.

m=7

$$T(n) = T\left(\frac{n}{7}\right) + n + T\left(\frac{5n}{7}\right)$$

Zu zeigen:

$$\exists\ c > 0 : T(n) \leq c \cdot n$$

Beweis:

$$T(n) \leq c \cdot \frac{n}{7} + n + c \cdot \frac{5n}{7} \leq c \cdot n$$

$$c = 7 \Leftrightarrow n + n + 5n \leq 7n \Rightarrow \exists\ c > 0 : T(n) \leq c \cdot n \Rightarrow T(n) \in O(n)$$

Somit ist die Laufzeit für $m = 7$ linear.

Nr.3

a)

x	Ergebnisse:
0	18
1	15
2	14
3	13
4	14
5	15
6	16
7	17
8	18
9	19
10	20
11	21
12	22
13	23

Für $x = x_2 = 3$, also dem Median der Menge $\{x_1, x_2, x_3\}$, wird die Summe minimal.

b)

Beh:
$\sum_{i=1}^{n} |x_i - x|$ wird minimal für $x = x_{k+1}$, d.h x := Median $\{x_1, ..., x_n\}$

Bew:
Sei die Menge aller x_i sortiert. Teile die Summe in zwei gleich große Teilsummen, wobei Summe 1 kleiner als x_{k+1} und Summe 2 größer als x_{k+1} ist:

$$\sum_{i=1}^{2k+1} |x_i - x|$$

$$= \sum_{i=1}^{k} |x_i - x| + \sum_{i=k+2}^{2k+1} |x_i - x| + |x_{k+1} - x|$$

$$= kx - \sum_{i=1}^{k} x_i - kx + \sum_{i=k+2}^{2k+1} x_i + |x_{k+1} - x|$$

$$= \sum_{i=k+2}^{2k+1} x_i - \sum_{i=1}^{k} x_i + |x_{k+1} - x|$$

<u>Für $x = x_{k+1}$:</u>

$$= \sum_{i=k+2}^{2k+1} x_i - \sum_{i=1}^{k} x_i$$

<u>Für $x \neq x_{k+1}$:</u>
Der Term $|x_{k+1} - x|$ wird immer > 0 und somit größer als der Fall $x = x_k$, d.h $\sum_{i=1}^{n} |x_i - x|$ ist genau dann minimal, wenn $x = x_{k+1}$ ist.

c)

Beh:
$\sum_{i=1}^{n} (x_i - x)^2$ wird minimal für $x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$, d.h x := arithmetisches Mittel von $\{x_1, ..., x_n\}$

Bew:

$$f(x) = \sum_{i=1}^{n} (x_i - x)^2$$

$$f'(x) = -\sum_{i=1}^{n} 2(x_i - x)$$

$$f''(x) = \sum_{i=1}^{n} 2$$

$\underline{f'(x) = 0:}$

$$
\begin{aligned}
&-\sum_{i=1}^{n} 2(x_i - x) = 0 \\
\Leftrightarrow\ &-2nx + \sum_{i=1}^{n} 2x_i = 0 \\
\Leftrightarrow\ &2\sum_{i=1}^{n} x_i = 2nx \\
\Leftrightarrow\ &\frac{1}{n} \cdot \sum_{i=1}^{n} x_i = x
\end{aligned}
$$

Da $f''(x)$ immer größer als Null ist, handelt es sich bei $x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$ um ein Minimum.
q.e.d