Übungen zur Vorlesung
Grundlagen der Programmierung II
Blatt 8

Aufgabe 1:

Ein Algorithmus habe folgende Laufzeit:

$$T(n) = \begin{cases} d, & \text{f r } n = 1 \\ T(n/2) + c * n, & \text{f r } n > 1 \end{cases} \qquad d, c > 0$$

Lösen Sie diese Rekursionsgleichung exakt auf, indem Sie zunächst eine Vermutung formulieren und diese dann durch vollständige Induktion beweisen.

Der Einfachheit halber darf angenommen werden, dass n eine Zweierpotenz ist.

Hinweis: $\sum_{i=0}^{n} x^i = \frac{x^{n+1} - 1}{x - 1}$

Aufgabe 2:

Zeigen Sie: Läßt man als einzige Operation zwischen Objekten Vergleiche zu benötigt man im schlimmsten Fall wenigsten n-1 Vergleiche, um zwei sortierte Folgen

$x_1 \leq x_2 \leq ... \leq x_{n/2}$

und $y_1 \leq y_2 \leq ... \leq y_{n/2}$

zu einer einzigen Folge $z_1 \leq z_2 \leq ... \leq z_n$ zu mischen.

Aufgabe 3:

Sei A eine Menge von Objekten mit einer Ordnungsrelation ≤. Ein Vorsortierungsmaß m: A*->IR ist eine Abbildung, die zu einer Folge $a_1,...,a_n$ den Grad der Vorsortierung als reelle Zahl bestimmt.

Beispiel für ein Vorsortierungsmaß ist die Funktion, die die Summe aller Abstände jedes Elements a_i von seinem in der sortierten Folge angenommen Platz ermittelt:

	7 12 4 3 8 6 9	->	3 4 6 7 8 9 12
Abstände:	3 5 1 3 0 3 1	->	Summe 16

a) Denken Sie sich zwei weitere Vorsortierungsmaße aus mit jeweils einem Beispiel aus.

b) Welches Vorsortierungsmaß besitzt eine aufsteigend sortierte, welches eine absteigend sortierte Folge?
c) Geben Sie jeweils eine Folge der Länge 6 an, die nach Ihren beiden Maßen maximal unsortiert ist.
d) Welche allgemeinen Eigenschaften sollten Vorsortierungsmaße besitzen?