Aufgabe 7.4 - Slowsearch

Mittwoch, 21. Juni 2017 18:04

a)

$T(n) = T(\frac{n}{2}) + cn$ Debugausgabe

$T(1) = \Theta 1$

Master Theorem: $cn + d \cdot r(n/b)$

$2 > 1$

$b > d$

$\Rightarrow \Theta(n)$

b) 1, Aus c) folgt, dass die Laufzeit im worst-case durch die Debugausgabe verschlechtert wird.

2, Pro Teilung des Arrays in zwei Teile besteht die Möglichkeit einer zusätzlichen Laufzeit von $O(n)$. Normalerweise benötigt ein Suchen einer Zahl durch die binäre Suche $O(\log n)$

c) $T(n) = T(\frac{n}{2}) + c$

$T(1) = \Theta 1$

Das Mastertheorem kann nicht verwendet werden, da bei c „n" fehlt.

Die Laufzeit muss linear gleich bleiben, größer werden oder schrumpfen.

Ohne Debugausgabe normale binäre Suche, d.h.

$T(n) \in \Theta(\log n)$

Induktion:

I.A.: $T(1) = \log_2(1) \cdot c + d = 0 \cdot c + d = d = \Theta 1$

I.V.: Angenommen es gilt, $T(n) \in \Theta(\log n)$ für beliebige, aber fixierte $n \in 2^i$ mit $i \in \mathbb{Z}$

I.S. $n \to 2n$

Zu beweisen: $T(2n) = c \log_2 n \,(2n) + d$

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

$$\log_2(n) \cdot c + d + e = \log_2(n) \cdot c + c + d = \left(\log_2(n) + \overbrace{\log_2(2)}^{1}\right) \cdot c + d$$

$$\log_2(2n) \cdot c + d$$ q.e.d.