Slowsort

Freitag, 16. Juni 2017 10:28

a)

T(n) = $T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + T\left(\left\lceil\frac{n}{2}\right\rceil\right) + T(n-1) + c$

T(0) = $\theta(1)$

obere Schranke:

Mit n >= 2 mit Ungleichung n-1 >= n/2

T(n-1) + T(n-1) +T(n-1) + c

T(n-1) auch dargestellt:

$$T(n) = 3^n + c * \sum_{i=0}^{n-1} 3i$$

$= 3^n + \frac{c}{2}(3^n - 1)$

Induktionsanfang:

T(0) = 1 + 0 = 1

Induktionsvorraussetzung:

T(n-1) = $3^n + \frac{c}{2}(3^n - 3 + 2)$

Induktionsschritt

T(n) = 3 * T(n-1) + c

$= 3 * \left(3^{n-1} + \frac{c}{2}(3^{n-1} - 1)\right) + c$

$= 3^n + \frac{3c}{2}(3^{n-1} - 1) + c$

$= 3^n + \frac{c}{2}(3^n - 3) + \frac{2c}{2}$

$= 3^n + \frac{c}{2}(3^n - 3 + 2)$

$= 3^n + \frac{c}{2}(3^n - 1)$

q.e.d

b)

Induktionsanfang:

Bei einem Element kommt man
sofort zum Rekursionsende (Zeile 2)
und ist somit fertig und hat ein
trivalerweise eine sortierte Liste

Induktionsschritt:

Für beliebiges, aber fixiertes n, gilt, dass
der Slowsort eine zufällig sortierte Liste
richtig sortiert

Induktionsschritt:

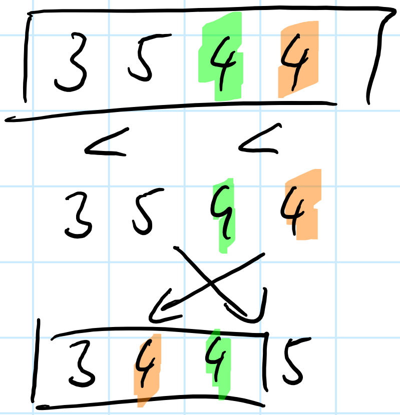
Jetzt betrachten wir eine zufällig sortierte
Liste mit n -> n+1.

1. Liste wird in Zeile 5 linksseitig geteilt, bis es nicht mehr geht
2. Nun können zweielementige oder mehrelementige rechtsteilig bzw. wieder linksteilig geteilt werden
3. Beginnend wird auf der linken Seite das Maximum nach rechts geschoben (nach m) und auf der rechten Seite das Maximum nach rechts geschoben (zum Ende)
4. Diese werden nun verglichen und das Maximum wird

ans Ende dieser ganzen Liste gebracht
5. Dies wird jetzt wieder wiederholt nur mit n-1 der Listen-elemente

c)
- Ist kein stabiler Algorithmus

Bsp.:



Bei einem stabilen Sortierverfahren behalten die Elemente, die gleich sind, ihre Position zueinander, was aber hier nicht der Fall ist: Die beiden Zahlen 4 tauschen ihre relative Positionen zueinander.