Laufzeit von QuickSort

Die Anzahl der Schritte hängt von der Wahl des Pivot-Elements ab:

Im Best Case wird bei jeden Rekursionsaufruf die Laufzeit halbiert:

$$T(n) = \underbrace{n}_{\text{Teile nach Pivot-Element: kleinere links, größere rechts, lösbar mit 1 for-schleife} \Rightarrow T=n} + \underbrace{T\left(\frac{n}{2}\right)}_{\text{Diride \& conquer: Wende den selben Algorithmus auf Liste Links und rechts vom Pivot an}} + T\left(\frac{n}{2}\right) = \underbrace{2\,T\left(\frac{n}{2}\right) + n}_{\text{Master-Methode: } a=2,\ b=2,\ f(n)=n} = \Theta(n \log n)$$

$\Rightarrow f(n) = n = \Theta(n^{\log_2 2}) = \Theta(n^0)$

$\Rightarrow$ 2. Fall

Im Worst Case wird bei jedem Rekursionsschritt nur ein Element bearbeitet:

$$T(n) = n + T(0) + T(n-1) = T(n-1) + n = \overbrace{n + n + \dots + n}^{n\text{-mal}} = \Theta(n^2)$$

Average Case Laufzeit von QuickSort

Zunächst stellen wir eine allgemeine Formel für die Laufzeit auf, wenn der Trennwert bei $i \in \{1, \dots, n\}$ liegt:

$$T_i(n) = T(i-1) + T(n-i) + n$$

Wenn man annimmt, dass alle Positionen gleich wahrscheinlich sind, ergibt sich:

$$T(n) = \frac{1}{n} \sum_{i=1}^{n} T_i(n) = \frac{1}{n} \sum_{i=1}^{n} \left(T(i-1) + T(n-i) + n \right)$$

Durchschnitt: Alle möglichen Laufzeiten aufsummiert geteilt durch die Anzahl der Möglichkeiten

$$= \frac{1}{n} \sum_{i=1}^{n} \left(T(i-1) + T(n-i) \right) + \underbrace{\frac{1}{n} \sum_{i=1}^{n} n}_{= \frac{1}{n} \cdot (\underbrace{n+n+\dots+n}_{n\text{-mal}}) = \frac{1}{n} \cdot n^2 = \frac{n^2}{n}}$$

Summe auseinanderziehen

$$= \frac{1}{n} \left(T(0) + T(n-1) + T(1) + T(n-2) + \dots + T(n-1) + T(0) \right) + \underbrace{\frac{n^2}{n}}_{=n}$$

Ändert man die Reihenfolge der Summanden, lässt sich ein Muster erkennen:

$$= \frac{1}{n} \left(T(0) + T(0) + T(1) + T(1) + \dots + T(n-1) + T(n-1) \right) + n$$

$$= \frac{2}{n} \sum_{i=0}^{n-1} T(i) + n$$

Wir eliminieren das $\frac{1}{n}$ durch Multiplikation mit n:

$$n \cdot T(n) = 2 \sum_{i=0}^{n-1} T(i) + n^2$$

Weiter gilt für $n-1$:

$$(n-1) \cdot T(n-1) = 2 \sum_{i=0}^{n-2} T(i) + (n-1)^2$$

Wenn man beide Gleichungen voneinander abzieht ergibt sich:

$$n \cdot T(n) - (n-1) \cdot T(n-1) = 2 \sum_{i=0}^{n-1} T(i) + n^2 - \left(2 \sum_{i=0}^{n-2} T(i) + \overbrace{(n-1)^2}^{n^2-2n+1} \right)$$

$$= 2 \sum_{i=0}^{n-1} T(i) - 2 \sum_{i=0}^{n-2} T(i) + n^2 - n^2 + 2n - 1 = \underbrace{2T(n-1)}_{\text{Die Summen sind bis auf den Summand } n-1 \text{ gleich, kürzen sich also fast vollständig weg.}} + 2n - 1$$

Insgesamt folgt:

$$n \cdot T(n) - (n-1) \cdot T(n-1) = 2T(n-1) + 2n - 1 \quad \Big| -2T(n-1)$$

$$\downarrow -(n-1) - 2 = -(n+1)$$

$$\Leftrightarrow \quad n \cdot T(n) - (n+1) \cdot T(n-1) = 2n - 1$$

Eine Teilung durch $n(n+1)$ liefert:

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

Und wenn man das $\frac{T(n-1)}{n}$ auf die andere Seite schiebt:

$$\Leftrightarrow \quad \frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2n-1}{n(n+1)}$$

Average Case Laufzeit von QuickSort – 2

Es gilt:

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

Wir substituieren $\frac{T(n)}{n-1}$ mit $\hat{T}(n)$ und landen bei einer Rekursionsgleichung:

$$\hat{T}(n) = \hat{T}(n-1) + \frac{2n-1}{n(n+1)}$$

Diese lässt sich mit Iterationsmethode lösen. Wir lösen auf, setzen also wiederholt ein:

$$\hat{T}(n) = \hat{T}(n-1) + \frac{2n-1}{n(n+1)}$$

$$= \hat{T}(n-2) + \frac{2(n-1)-1}{(n-1)((n-1)+1)} + \frac{2n-1}{n(n+1)} = \dots$$

$$= \hat{T}(n-i) + \sum_{k=0}^{i-1} \frac{2(n-k)-1}{(n-k)((n-k)+1)} = \hat{T}(n-i) + \sum_{k=0}^{i-1} \frac{2n-2k-1}{(n-k)(n-k+1)}$$

Mit Rekursionsbasis $n-i=1 \Rightarrow i = n-1$ folgt:

$$\hat{T}(1) + \sum_{k=0}^{n-2} \frac{2n-2k-1}{(n-k)(n-k+1)} = \hat{T}(1) + \sum_{i=2}^{n} \frac{2i-1}{i(i+1)}$$

Durch Ausrollen der Summe lässt sich eine einfachere Formel finden, NR:

$$\sum_{k=0}^{n-2} \frac{2n-2k-1}{(n-k)(n-k+1)} = \frac{2n-1}{n(n+1)} + \dots + \underbrace{\frac{\overbrace{2n-2(n-3)-1}^{5}}{\underbrace{(n-(n-3))(n-(n-3)+1)}_{12}}}_{} + \underbrace{\frac{\overbrace{2n-2(n-2)-1}^{3}}{\underbrace{(n-(n-2))(n-(n-2)+1)}_{6}}}_{} = \sum_{i=2}^{n} \frac{2i-1}{i(i+1)}$$

In der Summe taucht ein Bruch auf, den man zu einer harmonischen Reihe umformen könnte.

$$= \underbrace{\hat{T}(1) + \sum_{i=2}^{n} \frac{3}{i+1}}_{\text{Ergibt sich wenn man von der harmonischen Reihe auf den Schritt davor zurückrechnen will}} - \underbrace{\sum_{i=2}^{n} \frac{1}{i}}_{\text{harmonische Reihe}}$$

Das Wachstum der harmonischen Reihe ist nämlich bekannt:

$$\sum_{k=1}^{n} \frac{1}{k} = \Theta(\log n)$$

Schreibt man jetzt die beiden Summen geschickt auf, kürzt sich viel:

$$\hat{T}(1) + \sum_{i=2}^{n} \frac{3}{i+1} - \sum_{i=2}^{n} \frac{1}{i} = T(1) + \overbrace{\frac{3}{3} - \underbrace{\frac{1}{2}}_{\text{übrig}}}^{=2/3} + \underbrace{\frac{3}{4} - \frac{1}{3}}_{=2/4} + \overbrace{\frac{3}{5} - \frac{1}{4}}^{=2/5} + \dots + \underbrace{\frac{3}{n+1}}_{\text{übrig}} - \frac{1}{n}$$

$$= \underbrace{\hat{T}(1) - \overbrace{\frac{1}{2}}^{\text{übrig}} + \frac{3}{n+1}}_{\Theta(1)} + \underbrace{2 \cdot \sum_{i=3}^{n} \frac{1}{i}}_{\Theta(\log n)} = \Theta(\log n)$$

Durch Rücksubstitution erhalten wir schließlich die Laufzeit von QuickSort:

$$\hat{T}(n) = \frac{T(n)}{n-1} \Rightarrow T(n) = \underbrace{\hat{T}(n)}_{\Theta(\log n)} \cdot \underbrace{(n-1)}_{\Theta(n)} = \Theta(n \log n) \quad \square$$