

Algorithmik Blatt 1 Aufgabe 1

Mtr.-Nr. 6329857

Universität Hamburg — 19. Oktober 2019

Zusammengearbeitet mit 7330980

Was macht der Algorithmus?

DoSomething(X, x, start, end) sucht in dem Array X im Bereich zwischen Position start und end nach dem Vorkommen des Elements, welches dem doppelten von x entspricht, und gibt bei Erfolg die Position zurück, ansonsten 0.

Dazu wird der zu durchsuchende Bereich des Arrays in drei Drittel geteilt und geprüft ob das gesuchte Element auf den inneren Grenzen der Bereiche — sprich am Ende des ersten Drittels oder am Ende des zweiten Drittels — liegt. Ist dies der Fall, ist das Ergebnis gefunden. Liegt das Gesuchte Element auf keiner dieser beiden Position, werden die drei Drittel jeweils nach dem gleichen Prinzip durchsucht, angefangen mit dem hinteren Drittel. Der Algorithmus bricht nicht ab, wenn ein Ergebnis gefunden wurde, sondern verarbeitet alle bereits geteilten Drittel — sowie deren weitere Zerteilungen — bis zu Ende.

Rekursive Form

```
1 DoSomething(X, x, start, end)
2     m_1 = floor(start + (end - start) / 3) // 1/3 Position
3     m_2 = floor(start + 2 * (end - start) / 3) // 2/3 Position
4     if (start > end) // Rekursionsabbruch
5         return 0
6     if (X[m_1] == 2 * x) // Element gefunden
7         return m_1
8     else if (X[m_2] == 2 * x) // Element gefunden
9         return m_2
10    else
11        return max( // Rekursion, nur einer kann >0 sein
12            DoSomething(X, x, m_2 + 1, end), // (R1)
13            DoSomething(X, x, m_1 + 1, m_2 - 1), // (R2)
14            DoSomething(X, x, start, m_1 - 1) // (R3)
15        )
```

Rekurrenzgleichung

Der nichtrekursive Anteil der Funktion besteht nur aus einfachen arithmetischen Operationen, benötigt also konstant c viel Zeit. Die Eingabegröße n ergibt sich aus der Differenz Parametern start und end , da diese den zu verarbeitenden Bereich angeben. Die Parameter X und x sind über die Rekursionstiefe konstant.

Der Algorithmus teilt das Problem in 3 Teilprobleme und reduziert die Problemgröße dabei von n auf $\frac{n}{3} - 1$ bzw. auf $\frac{n}{3} - 2$. Denn sei $n = \text{end} - \text{start}$ ergibt sich:

- für Aufruf R_1 : $n = \text{end} - (m_2 + 1) = (\text{end} - \text{start})/3 - 1$
- für Aufruf R_2 : $n = (m_2 - 1) - (m_1 + 1) = (\text{end} - \text{start})/3 - 2$
- und für Aufruf R_3 : $n = (m_1 - 1 - \text{start}) = (\text{end} - \text{start})/3 - 1$.

Die Abbruchbedingung ist dass $\text{start} > \text{end}$, also dass $n < 0$ ist.

Daraus ergibt sich die Rekurrenzgleichung:

$$\begin{aligned} T(n) &= c && \text{wenn } n < 0 \\ T(n) &= 3 \cdot T(n/3 - q) + c && \text{wenn } n \geq 0 \end{aligned}$$

Wobei q je nach Rekursionschritt 1 oder 2 ist, was aber, wie wir später sehen werden, keinen Unterschied macht.

Um das Mastertheorem verwenden zu können, bringen wir die Gleichung in die richtige Form:

$$\begin{aligned} \text{Sei } l &= n - 1 \\ T(l) &= c && \text{wenn } n = 0 \\ T(l) &= 3 \cdot T(l/3 - q) + c && \text{wenn } l > 0 \\ T(l) &= c && \text{wenn } l = 0 \\ T(l) &= 3 \cdot T((l - 3 \cdot q)/3) + c && \text{wenn } l > 0 \end{aligned}$$

$$\begin{aligned} \text{Sei } m &= l - 3 \cdot q \\ T(m + 3 \cdot q) &= 3 \cdot T(m/3) + c \\ S(m) &= T(m + 3 \cdot q) \\ S(m) &= 3 \cdot S(m/3) + c \end{aligned}$$

Nun hat die Rekurrenzgleichung die Passende Form, um das Mastertheorem (MT) anwenden zu können. Da die Parameter a und b des Mastertheorems 3 sind und der nichtrekursive Teil konstant ist, tritt Fall 1 des MT ein, wonach der rekursive Anteil dominiert und sich

$$S(m) = \theta(N^{\log_3 3}) = \theta(m^{\log_3 3}) = \theta(m) = \theta(l - (3 \cdot q)) = \theta(n - 1 - (3 \cdot q)) = \theta(n)$$

ergibt. Bisher hatten wir keinen Wert für q gewählt, doch nun hat sich sowieso ergeben, dass er nur als konstanter Summand auftritt, also im O -Kalkül vernachlässigt werden kann.

Substitutionsmethode

$$T(N, M) = \begin{cases} 1 & \text{für } N = 1 \\ T(N-1, M) + N & \text{für } 1 < N \leq M \\ 2 \cdot T\left(\frac{N}{2}, M\right) + N & \text{für } N > M \end{cases}$$

Für $1 < N \leq M$

$$\begin{aligned} T(N, M) &= T(N-1, M) + N \\ &= T(N-2, M) + N + N \\ &= T(N-(N-1), M) + (N-1) \cdot N \\ &= T(1) + N^2 - N \\ &= 1 + N^2 - N \end{aligned}$$

Für $N > M$

$$\begin{aligned} T(N, M) &= 2 \cdot T\left(\frac{N}{2}, M\right) + N \\ &= 2 \cdot \left(2 \cdot T\left(\frac{N}{2}, M\right) + N\right) + N \\ &= 4 \cdot T\left(\frac{N}{4}, M\right) + 2N + N \\ &= 2^i \cdot T\left(\frac{N}{2^i}, M\right) + 2^{i-1}N + \dots + N^1 \\ &= 2^i \cdot T\left(\frac{N}{2^i}, M\right) + \sum_{k=0}^{i-1} 2^k N \\ i &\geq \log_2\left(\frac{N}{M}\right) \text{ damit } \frac{N}{2^i} \leq M \\ i &= \log_2\left(\frac{N}{M}\right) \text{ weil } N, M \text{ Zweierpotenz} \\ T(N, M) &= 2^{\log_2(\frac{N}{M})} \cdot T\left(\frac{N}{2^{\log_2(\frac{N}{M})}}, M\right) + \sum_{k=0}^{\log_2 \frac{N}{M} - 1} 2^k N \\ &= \frac{N}{M} \cdot T\left(\frac{N}{M}, M\right) + \sum_{k=0}^{\log_2(\frac{N}{M}) - 1} 2^k N \\ &= \frac{N}{M} \cdot T(M, M) + \sum_{k=0}^{\log_2 \frac{N}{M} - 1} 2^k N \\ &= \frac{N}{M} \cdot T(M, M) + \sum_{k=0}^{\log_2(\frac{N}{M}) - 1} 2^k N \\ &= \frac{N}{M} \cdot T(M, M) + N \sum_{k=0}^{\log_2 \frac{N}{M} - 1} 2^k \end{aligned}$$

Die Summe ist eine geometrische Reihe, also:

$$\begin{aligned}
T(N, M) &= \frac{N}{M} \cdot T(M, M) + N \left(\frac{2^{\log_2 \frac{N}{M}} - 1}{1} \right) \\
&= \frac{N}{M} \cdot T(M, M) + N \left(\frac{N}{M} - 1 \right) \\
&= \frac{N}{M} \cdot T(M, M) + \left(\frac{N^2}{M} - N \right)
\end{aligned}$$

Ersetzen $T(M, M)$ durch $T(N, M)$ für $1 < N \leq M$.

$$\begin{aligned}
T(N, M) &= \frac{N}{M} \cdot (1 + M^2 - M) + \left(\frac{N^2}{M} - N \right) \\
&= \left(\frac{N}{M} + \frac{N}{M} \cdot M^2 - M \cdot \frac{N}{M} \right) + \left(\frac{N^2}{M} - N \right) \\
&= \left(\frac{N}{M} + NM - N \right) + \left(\frac{N^2}{M} - N \right) \\
&= \frac{N^2}{M} + \frac{N}{M} + NM - 2N \\
&= \frac{N^2 + N}{M} + N(M - 2)
\end{aligned}$$