Theoretische Informatik 1

Gruppe 5 (Maik Weinard)

3. Übungsblatt

Adam Duck

7. November 2001

3.1 Rekursionsgleichungen 14 (4+4+6) Wertungspunkte

Löse die folgenden Rekursionsgleichungen. (Um die Berechnung zu vereinfachen, ignorieren wir den Einfluss von Rundungsoperationen. Wir stellen uns die Funktionen T also als Funktionen reeller Zahlen vor und wenden den Basisfall an, sobald das Argument kleinergleich der genannten Schranke ist.)

(a) $T(n) = \sqrt{n} \cdot T(\sqrt{n}) + n, \qquad T(2) = 1$ (4 Punkte)

Lösung:

$$\begin{aligned}
T(n) &= n^{\frac{1}{2}} \cdot T\left(n^{\frac{1}{2}}\right) + n \\
&= n^{\frac{1}{2}}\left(n^{\frac{1}{4}} T\left(n^{\frac{1}{4}}\right) + n^{\frac{1}{2}}\right) + n &&= n^{\frac{3}{4}} T\left(n^{\frac{1}{4}}\right) + 2n \\
&= n^{\frac{3}{4}}\left(n^{\frac{1}{8}} T\left(n^{\frac{1}{8}}\right) + n^{\frac{1}{4}}\right) + 2n &&= n^{\frac{7}{8}} T\left(n^{\frac{1}{8}}\right) + 3n \\
&= n^{\frac{7}{8}}\left(n^{\frac{1}{16}} T\left(n^{\frac{1}{16}}\right) + n^{\frac{1}{8}}\right) + 3n &&= n^{\frac{15}{16}} T\left(n^{\frac{1}{16}}\right) + 4n \\
&\vdots \\
&= n^{\frac{2^k-1}{2^k}} \cdot T\left(n^{\frac{1}{2^k}}\right) + kn
\end{aligned}$$

Diese Vermutung wollen wir nun mittels Vollständiger Induktion beweisen:

IV $k = 0$

$$\begin{aligned}
T(n) &\stackrel{?}{=} n^{\frac{2^0-1}{2^0}} \cdot T\left(n^{\frac{1}{2^0}}\right) + 0n \\
&= T(n) \qquad \checkmark
\end{aligned}$$

IS $k \to k+1$

$$\begin{aligned}
T(n) &= n^{\frac{2^k-1}{2^k}} \cdot T\left(n^{\frac{1}{2^k}}\right) + kn \\
&= n^{\frac{2^k-1}{2^k}} \cdot \left(\left(n^{\frac{1}{2^k}}\right)^{\frac{1}{2}} \cdot T\left(\left(n^{\frac{1}{2^k}}\right)^{\frac{1}{2}}\right) + n^{\frac{1}{2^k}}\right) + kn \\
&= n^{\frac{2^{k+1}-2}{2^{k+1}}} \cdot \left(n^{\frac{1}{2^{k+1}}} \cdot T\left(n^{\frac{1}{2^{k+1}}}\right) + n^{\frac{2}{2^{k+1}}}\right) + kn \\
&= n^{\frac{2^{k+1}-1}{2^{k+1}}} \cdot T\left(n^{\frac{1}{2^{k+1}}}\right) + (k+1)n
\end{aligned}$$

Womit die Vermutung gezeigt wäre.

Jetzt brauchen wir nur noch den Basisfall zu bestimmen, hier gilt:

$$\begin{aligned} n^{\frac{1}{2^k}} &= 2 \\ n &= 2^{(2^k)} \\ k &= \log_2 \log_2 n \end{aligned} \tag{1.1}$$

Somit

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

Wir können noch – zur Sicherheit – dieses Ergebnis für denn Fall $n = 2$ überprüfen:

$$\begin{aligned} T(2) &= 2^{\frac{\log_2 2 - 1}{\log_2 2}} + 2 \log_2 \log_2 2 \\ &= 2^{\frac{0}{1}} + 2 \cdot 0 = 1 \end{aligned}$$

(b) $T(n) = 2\sqrt{n} \cdot T(\sqrt{n}) + n, \qquad T(2) = 1$ (4 Punkte)

Lösung:

$$\begin{aligned} T(n) &= 2n^{\frac{1}{2}} \cdot T\left(n^{\frac{1}{2}}\right) + n \\ &= 2n^{\frac{1}{2}}\left(2n^{\frac{1}{4}} T\left(n^{\frac{1}{4}}\right) + n^{\frac{1}{2}}\right) + n &&= 4n^{\frac{3}{4}} T\left(n^{\frac{1}{4}}\right) + 3n \\ &= 4n^{\frac{3}{4}}\left(2n^{\frac{1}{8}} T\left(n^{\frac{1}{8}}\right) + n^{\frac{1}{4}}\right) + 3n &&= 8n^{\frac{7}{8}} T\left(n^{\frac{1}{8}}\right) + 7n \\ &= 8n^{\frac{7}{8}}\left(2n^{\frac{1}{16}} T\left(n^{\frac{1}{16}}\right) + n^{\frac{1}{8}}\right) + 7n &&= 16n^{\frac{15}{16}} T\left(n^{\frac{1}{16}}\right) + 15n \\ &\vdots \\ &= 2^k n^{\frac{2^k-1}{2^k}} \cdot T\left(n^{\frac{1}{2^k}}\right) + (2^k - 1)n \end{aligned}$$

Auch diese Vermutung muss bewiesen werden:

IV $k = 0$

$$\begin{aligned} T(n) &\overset{?}{=} 2^0 n^{\frac{2^0-1}{2^0}} \cdot T\left(n^{\frac{1}{2^0}}\right) + (2^0 - 1)n \\ &= T(n) \qquad \checkmark \end{aligned}$$

IS $k \to k + 1$

$$\begin{aligned} T(n) &= 2^k n^{\frac{2^k-1}{2^k}} \cdot T\left(n^{\frac{1}{2^k}}\right) + (2^k - 1)n \\ &= 2^k n^{\frac{2^k-1}{2^k}} \cdot \left(2\left(n^{\frac{1}{2^k}}\right)^{\frac{1}{2}} \cdot T\left(\left(n^{\frac{1}{2^k}}\right)^{\frac{1}{2}}\right) + n^{\frac{1}{2^k}}\right) + (2^k - 1)n \\ &= 2^k n^{\frac{2^{k+1}-2}{2^{k+1}}} \cdot \left(2n^{\frac{1}{2^{k+1}}} \cdot T\left(n^{\frac{1}{2^{k+1}}}\right) + n^{\frac{2}{2^{k+1}}}\right) + (2^k - 1)n \\ &= 2^{k+1} n^{\frac{2^{k+1}-1}{2^{k+1}}} \cdot T\left(n^{\frac{1}{2^{k+1}}}\right) + 2^k n + (2^k - 1)n \\ &= 2^{k+1} n^{\frac{2^{k+1}-1}{2^{k+1}}} \cdot T\left(n^{\frac{1}{2^{k+1}}}\right) + (2^{k+1} - 1)n \end{aligned}$$

Womit die Vermutung gezeigt wäre.

Jetzt ist $T(n)$ nur noch auf den Basisfall zurückzuführen, wobei hier dieselbe Rechnung wie in (1.1) zu folgendem Ergebnis führt:

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

(c) $T(n) = 2T(n/2) + n \log_2 n, \qquad T(1) = 1$ (6 Punkte)

Lösung:

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right) + n\log_2 n \\ &= 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\log_2 \frac{n}{2}\right) + n \log_2 n \\ &= 4T\left(\frac{n}{4}\right) + n\log_2 \frac{n}{2} + n\log_2 n \\ &= 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\log_2 \frac{n}{4}\right) + n\log_2 n + n\log_2 \frac{n}{2} \\ &= 8T\left(\frac{n}{8}\right) + n\log_2 n + n\log_2 \frac{n}{2} + n\log_2 \frac{n}{4} \\ &= 8\left(2T\left(\frac{n}{16}\right) + \frac{n}{8}\log_2 \frac{n}{8}\right) + n\log_2 n + n\log_2 \frac{n}{2} + n\log_2 \frac{n}{4} \\ &= 16T\left(\frac{n}{16}\right) + n\log_2 n + n\log_2 \frac{n}{2} + n\log_2 \frac{n}{4} + n\log_2 \frac{n}{8} \\ &\vdots \\ &= 2^k T\left(\frac{n}{2^k}\right) + n\sum_{i=0}^{k-1} \log_2\left(\frac{n}{2^i}\right) \\ &= 2^k T\left(\frac{n}{2^k}\right) + n\left(\sum_{i=0}^{k-1} \log_2 n - \sum_{i=0}^{k-1} \log_2 2^i\right) \\ &= 2^k T\left(\frac{n}{2^k}\right) + n\left(k \log_2 n - \frac{k(k-1)}{2}\right) \end{aligned}$$

Auch hier folgt ein Beweis mittels Vollständiger Induktion.

IV $k = 0$

$$\begin{aligned} T(n) &\stackrel{?}{=} 2^0 T\left(\tfrac{n}{2^0}\right) + n\left(0 \log_2 n - \tfrac{0(0-1)}{2}\right) \\ &= T(n) \qquad \checkmark \end{aligned}$$

IS $k \to k+1$

$$\begin{aligned} T(n) &= 2^k T\left(\frac{n}{2^k}\right) + kn\log_2 n - \frac{n}{2}k(k-1) \\ &= 2^k\left(2T\left(\frac{n}{2^k \cdot 2}\right) + \frac{n}{2^k}\log_2\left(\frac{n}{2^k}\right)\right) + kn\log_2 n - \frac{n}{2}k(k-1) \\ &= 2^{k+1} \cdot T\left(\frac{n}{2^{k+1}}\right) + n\log_2 n - n\log_2\left(2^k\right) + kn\log_2 n - \frac{n}{2}k(k-1) \\ &= 2^{k+1} \cdot T\left(\frac{n}{2^{k+1}}\right) + (k+1)n\log_2 n - \frac{n}{2}\left(\frac{2k + k(k-1)}{2}\right) \\ &= 2^{k+1} \cdot T\left(\frac{n}{2^{k+1}}\right) + (k+1)n\log_2 n - n\frac{(k+1)((k+1)-1)}{2} \end{aligned}$$

Womit die Vermutung gezeigt wäre.

Im Basisfall muss gelten:

$$\begin{aligned}\frac{n}{2^k} &= 1\\ k &= \log_2 n\end{aligned}$$

Eingesetzt in die – nun bewiesene – "Vermutung":

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

Hinweis: Entwickle, wie im Beweis vom Mastertheorem, die Rekursionsgleichung, bilde eine Vermutung, beweise die Vermutung und führe auf dem Basisfall zurück. (Achtung, das Mastertheorem ist hier nicht mehr anwendbar.)

3.2 Elementare Datenstrukturen 10 Wertungspunkte

Entwerfe eine Datenstruktur, die Zahlen speichert und die die Operationen `pop`, `push(x)` und `minfind` in **konstanter Zeit** unterstützt: Die Dauer einer jeden Operation darf also nicht von der Anzahl der schon in der Datenstruktur enthaltenen Zahlen abhängen.

Zur Erinnerung: `pop` entfernt die zuletzt eingefügte Zahl aus der Datenstruktur, `push(x)` fügt die Zahl `x` in die Datenstruktur ein. Die Funktion `minfind` benennt den Wert der kleinsten gespeicherten Zahl, verändert aber die Menge der gespeicherten Zahlen nicht. Beachte, dass die Datenstruktur Elemente auch mehrfach enthalten kann. (10 Punkte)

Lösung:

Das machen wir am besten mit einem Stack, wobei jedes Objekt (`O`) auf dem Stack 2 Elemente hat. `O` ist also ein Array mit 2 Elementen. Das erste Element (`E1`) des Arrays `O` auf dem Stack enthält die eigentliche Zahl, die wir speichern sollen. Das zweite Element (`E2`) enthält immer das derzeitige Minimum – kann also auch dieselbe Zahl enthalten wie `E1`.

Die Funktionen `push(x)`, `pop` und `minfind` werden wie folgt definiert:

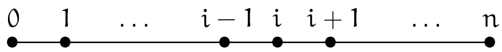
```
push(x) {
   Setze temp = pop
   Lege temp wieder auf den Stack zurück
   wenn x < temp[1]
      Lege (x,x) auf den Stack
   sonst
      Lege (x,temp[1]) auf den Stack
}

pop {
   temp = hole oberstes Element vom Stack
   Gib temp[0] zurück
}
```

```
minfind {
   temp = hole oberstes Element vom Stack
   Lege temp wieder auf den Stack zurück
   Gib temp[1] zurück
}
```

Wie man leicht sehen kann, hängt die Laufzeit der Funktionen nicht von der Eingabelänge ab. Wenn wir den Stack mit Elementen auffüllen, tragen wir immer das derzeitige kleinste Element mit uns. Wenn jetzt das kleinste Objekt auf dem Stack durch `pop` gelöscht wird, haben wir im nächsten Objekt auf dem Stack das nächstkleinere Minimum.

3.3 Random Walk keine Wertung

In einem Teich befinden sich $n+1$ Steine $0, 1, \ldots, n$ in einer Reihe. Ein Frosch sitzt anfänglich auf Stein n. Befindet er sich zu irgend einem Zeitpunkt auf einem Stein, der einen linken und einen rechten Nachbarstein hat, so springt er jeweils mit Wahrscheinlichkeit $1/2$ auf einen der beiden. Erreicht er eines der Enden der Reihe, so springt er im nächsten Sprung mit Sicherheit auf den einzigen Nachbarn. Nach wievielen Sprüngen wird der Frosch erwartungsgemäß zum ersten mal Stein 0 erreichen?



Sei $E_n(i)$ die erwartete Zahl der Sprünge von Stein i aus bis zum erstmaligen Erreichen von Stein 0. Wir können die Erwartungswerte kanonisch in Abhängigkeit voneinander formulieren.

$$
\begin{aligned}
E_n(0) &= 0 \\
E_n(i) &= 1 + \frac{1}{2} \cdot E_n(i-1) + \frac{1}{2} \cdot E_n(i+1) && \text{für } 1 \leq i \leq n-1 \\
E_n(n) &= 1 + E_n(n-1)
\end{aligned}
$$

Berechne $E_n(n)$.