46. (0 Punkte) Beweisen Sie folgende Gleichungen für Listen `x` und `y` durch strukturelle Rekursion aus den unten angegebenen Definitionen.

$$\begin{aligned} \texttt{x ++ [ ]} &= \texttt{x} & (1)\\ \texttt{length (x ++ y)} &= \texttt{(length x) + (length y)} & (2)\\ \texttt{length (reverse x)} &= \texttt{(length x)} & (3)\\ \texttt{reverse (x ++ y)} &= \texttt{(reverse y) ++ (reverse x)} & (4)\\ \texttt{reverse (reverse x)} &= \texttt{x} & (5) \end{aligned}$$

```
[ ] ++ yl = yl
(x : xr) ++ yl = x :(xr ++ yl)
reverse [ ] = [ ]
reverse (x : xr) = (reverse xr) ++ [x]
```

```
length [ ] = 0
length (x : xr) = 1 + (length xr)
```

47. (0 Punkte) Beweisen Sie (zum Beispiel durch strukturelle Induktion), dass jeder vollständig balancierten ternäre Baum der Höhe h genau $(3^{h+1} - 1)/2$ Knoten enthält. (Jeder innere Knoten hat genau 3 Kinder, und alle Blätter befinden sich auf Tiefe h.)

48. (0 Punkte) Ein binärer Suchbaum soll Einträge speichern, die außer dem Schlüssel x_i noch ein „Gewicht" w_i enthalten. Die Algorithmen zum Suchen, Einfügen und Löschen in binären Suchbäumen sollen so erweitert werden, dass man jederzeit zu zwei Werten a und b das Gesamtgewicht aller Einträge finden kann, deren Schlüssel x_i im Intervall $a < x_i \le b$ liegt. Die Laufzeit für alle Operationen soll höchstens proportional zur Höhe des Baumes sein.

    Beschreiben Sie die Felder, die in den Knoten enthalten sind (zusammen mit einer Erklärung der Bedeutung, sofern sie nicht offensichtlich ist), und schreiben Sie ein Programmstück (Methoden in Java oder Funktionen in Haskell) für die oben beschriebene Intervallabfrage und zum Einfügen eines neuen Elementes.

49. (8 Punkte) Spezifizieren Sie den abstrakten Datentyp einer Prioritätswarteschlange, die mindestens die Operationen `entferneMin`, `einfügen` und `istLeer` enthält. Die Prioritätswarteschlange soll *Einträge* (x_i, a_i) enthalten, die einen *Schlüssel* x_i (eine Zahl) und einen *Wert* a_i enthalten. (Entscheiden Sie, wie sie mit gleichen Schlüsseln umgehen wollen.)

50. (12 Punkte) Modifizieren Sie das Programm zur Warteschlangensimulation aus Aufgabe 22 vom 3. Übungsblatt, Variante (a) folgendermaßen: Die Ankunftsrate der Patienten ist $\lambda = 5/\mathrm{h}$, und es gibt $n = 3$ Ärztinnen. Es gibt eine zusätzliche Kategorie von *Notfallpatienten* mit Ankunftsrate $\lambda = 0{,}5/\mathrm{h}$. Bei Ankunft eines Notfallpatienten unterbrechen alle n Ärztinnen ihre augenblickliche Tätigkeit und beschäftigen sich für einen Zeitraum, der gleichverteilt im Intervall $[a \,..\, b]$ mit $a = 30\,\mathrm{min}$ und $b = 40\,\mathrm{min}$ ist, mit dem neuen Patienten. Danach wird der Notfallpatient zu einer anderen Abteilung geschickt und verlässt das hier untersuchte System, und die Ärztinnen fahren mit der vorher unterbrochenen Tätigkeit fort. Ein neuer Notfallpatient, der während der Behandlung eines Notfallspatienten eintrifft, kommt dran, wenn diese Behandlung abgeschlossen ist.

51. (0 Punkte) Bestimmen des doppelten Elementes.[1]
    In einem Array $a_0, a_1, \ldots, a_n$ sind ganzzahlige Werte zwischen 1 und n gespeichert. Daraus folgt, dass es mindestens eine Zahl geben muss, die doppelt vorkommt. Schreiben Sie ein Programm, das eine dieser Zahlen bestimmt. Das Programm soll lineare Laufzeit haben, auf das Array a nur *lesend* zugreifen, und nur konstanten zusätzlichen Speicher benötigen.

[1] Diese Aufgabe dient zur Entspannung und hat mit dem Stoff der Vorlesung nichts zu tun. Wenn Sie das Rätsel aus einer anderen Quelle bereits kennen, dann würde mich sehr interessieren, woher.