20. (2 Punkte) Beweisen Sie, dass man zum Finden eines Elementes in einer sortierten Liste mit n Elementen mindestens $\Omega(\log n)$ Vergleiche benötigt.

21. (a) (3 Punkte) Schreiben Sie eine Schnittstelle (`interface`) oder eine abstrakte Klasse `PWSchlange` für eine Prioritätswarteschlange, die mindestens die beiden Methoden `entferneMin` und `einfügen` enthält. Die Prioritätswarteschlange soll Objekte vom Typ `Comparable` verwalten.

    (b) (3 Punkte) Schreiben Sie eine Klasse `Halde`, die eine `PWSchlange` implementiert beziehungsweise erweitert![1]

22. (12 Punkte) Simulation einer Warteschlange in Java.
In einer Unfallambulanz müssen sich die ankommenden Patienten zunächst anmelden. Danach werden Sie in der Reihenfolge der Anmeldung von einer der $n = 2$ diensthabenden Ärztinnen befragt und untersucht. Gegebenfalls wird ein Röntgenbild angefertigt, und anschließend werden die Patienten behandelt.
Die Ankunft der Patienten ist ein Poisson-Prozess mit Rate $\lambda = 2\,\mathrm{h}^{-1}$. Das heißt, dass der Abstand von der Ankunft eines Patienten bis zur Ankunft des nächsten Patienten exponentialverteilt mit dem Mittelwert $1/\lambda = 30$ Minuten ist. Eine exponentialverteilte Zufallsvariable mit Mittelwert μ kann man mit der Formel `-`μ`*Math.log(Math.random())` erzeugen. Die Zeit, die zur Anmeldung nötig ist, soll vernachlässigt werden. Die Dauer der ersten Untersuchung ist gleichverteilt im Intervall $[a\,..\,b]$ mit $a = 5\,\mathrm{min}$ und $b = 20\,\mathrm{min}$. Eine solche gleichverteilte Zufallsvariable kann man mit der Formel `a+(b-a)*Math.random()` erzeugen. Anschließend ist mit Wahrscheinlichkeit $p = 0{,}15$ die Behandlung beendet, und der Patient wird entlassen. Andernfalls wird der Patient zum Röntgen geschickt. Die Patienten werden dort in der Reihenfolge ihrer Ankunft an der Röntgenstation drangenommen. Die Zeit zur Erstellung eines Röntgenbildes ist gleichverteilt im Intervall $[a\,..\,b]$ mit $a = 5\,\mathrm{min}$ und $b = 10\,\mathrm{min}$ (unabhängig von der Dauer der Untersuchung). Die anschließende Behandlung ist wieder eine unabhängige Zufallsvariable im Intervall $[a\,..\,b]$ mit $a = 10\,\mathrm{min}$ und $b = 30\,\mathrm{min}$.
Untersuchen Sie nun folgende Varianten:

    (a) Jeder neue Patient wird der ersten freiwerdenden Ärztin zugeteilt. Nach dem Fertigstellen des Röntgenbildes muss der Patient von derselben Ärztin behandelt werden. Wenn eine Ärztin frei wird, nimmt sie als nächstes bevorzugt einen Patienten an, der von der Röntgenabteilung zurückkommt.

    (b) Wie oben, aber die Patienten, die vom Röntgen zurückkommen, reihen sich in dieselbe Schlange ein wie die neuen Patienten.

    Simulieren Sie diese Schlange von 7:00 bis 11:00 Uhr eines Tages. (Um 11:00 Uhr wird die Ambulanz geschlossen; es werden keine weiteren Patienten mehr angenommen, aber alle Patienten, die sich schon angemeldet haben, werden noch behandelt.)
    Bestimmen Sie für jede Variante die durchschnittliche Wartezeit eines Patienten, die maximale Wartezeit, und die Gesamtzeit, wie lange die Ärztinnen nichts zu tun hatten (von 7:00 Uhr bis zur Behandlung des jeweils letzten Patienten). Führen Sie je drei unabhängige Simulationsläufe durch.
    Schreiben Sie Ihr Programm so, dass es möglichst leicht zu ändern oder zu erweitern ist (zum Beispiel für unterschiedliche Werte von λ, p und n). Verwenden Sie ein Objekt vom Typ `PWSchlange` aus der vorigen Aufgabe zur Verwaltung der nächsten Ereignisse.
    Stellen Sie die Hierarchie aller Klassen und Schnittstellen, die Sie definiert haben, dar.

[1] siehe das Programm aus der Vorlesung: `http://www.inf.fu-berlin.de/~rote/Lere/2003-04-WS/Algorithmen+Programmierung3/heapsort.java`