Algorithmen und Programmieren 3, WS 1999/2000 — 3. Übungsblatt

Abgabe bis Dienstag, 16. November 1999

17. (7 Punkte)

    (a) 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) Schreiben Sie eine Klasse `Halde`, die eine `PWSchlange` implementiert beziehungsweise erweitert.[1]

18. (13 Punkte) Simulieren Sie eine einfache Warteschlange in Java.

    In einem Supermarkt mit $n = 3$ Kassen stellen sich ankommende Kunden bei einer der Kassen an, bis sie an dieser Kasse bedient werden.

    Die Ankunft der Kunden ist ein Poisson-Prozess mit Rate $\lambda = 1{,}5\,\text{min}^{-1}$. Das heißt, dass der Abstand von der Ankunft eines Kunden bis zur Ankunft des nächsten Kunden exponentialverteilt mit dem Mittelwert $1/\lambda = 40$ Sekunden ist. Eine exponentialverteilte Zufallsvariable mit Mittelwert μ kann man mit der Formel `-`μ`*Math.log(Math.random())` erzeugen.

    Die Zeit zur Bedienung eines Kunden an der Kasse ist gleichverteilt im Intervall $[a \,..\, b]$ mit $a = 10\,\text{sec}$ und $b = 200\,\text{sec}$. Eine solche gleichverteilte Zufallsvariable kann man mit der Formel `a+(b-a)*Math.random()` erzeugen.

    Untersuchen Sie nun folgende Fälle:

    (a) Es gibt eine gemeinsame Schlange für alle Kassen.

    (b) Es gibt eine getrennte Schlange für jede Kasse. Neu ankommende Kunden stellen sich bei einer zufälligen Kasse an, (auch wenn eine andere Kasse frei wäre), und bleiben dort stehen, bis sie bedient werden.

    (c) (freiwillige Zusatzaufgabe, 1 Punkt.) Es gibt eine getrennte Schlange für jede Kasse. Neu ankommende Kunden stellen sich bei der kürzesten Schlange an. Wenn mehrere Schlangen die gleiche Länge haben, dann wird die „erste“ Schlange (mit der kleinsten Nummer) gewählt.

    Simulieren Sie diese Schlange von 9:00 bis 19:00 Uhr eines Tages. (Um 19:00 Uhr wird der Eingang geschlossen; es stellen sich keine weiteren Kunden mehr an, aber alle Kunden, die sich schon angestellt haben, werden noch bedient.) Bestimmen Sie für jede Variante die durchschnittliche Wartezeit eines Kunden, die maximale Wartezeit, und die Gesamtzeit, wie lange die Kassiererinnen nichts zu tun hatten. 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 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 die Programme aus der Vorlesung, `http://www.inf.fu-berlin.de/~rote/Lere/alp3/heapsort.java`