Universität
des
Saarlandes
FR 6.2 Informatik

Prof. Dr. Kurt Mehlhorn, Dr. Konstantinos Panagiotou **WiSe 2011/2012**

Übungen zu Computational Thinking

http://www.mpi-inf.mpg.de/departments/d1/teaching/ws11/ct/

Blatt 9 Abgabeschluss: 16.1.12 16:00

Regeln: Bis zum Semesterende müssen mindestens 42% der maximal erreichbaren Punkte aller Übungszettel erworben werden.
Programmcode ist elektronisch per E-Mail abzugeben. Zusätzlich müssen die Ausgaben einer exemplarischen Programmausführung mitgeliefert werden.

Bleistiftaufgaben

Aufgabe 1 (*40 Punkte*) Aufgaben eines Betriebssystems: Sie sollen eine kleine Firma organisieren. Sie haben drei Arbeiter, 2 Schaufeln und 2 Schubkarren. Sie erhalten Aufträge von der Form

$$(x \text{ Arbeiter}, y \text{ Schaufeln}, z \text{ Schubkarren}, u \text{ Minuten}).$$

Die Erledigung eines solchen Auftrags braucht x Arbeiter, y Schaufeln, z Schubkarren, und u Minuten. Es ist stets $x \leq 3$, $y \leq 2$ und $z \leq 2$.

- Nehmen Sie zunächst an, dass ihre Arbeiter absolut zuverlässig sind. Sie bringen Werkzeug zuverlässig zurück und brauchen genau die vorgegebene Zeit. Wir würden Sie die Firma organisieren? Über welche Größen würden Sie buchführen? Etwa, Anzahl der Arbeiter, die im Augenblick nicht beschäftigt sind, Anzahl der noch verfügbaren Schaufeln, Menge der Aufgaben, die gerade bearbeitet werden uns seit wann. Menge der unerledigten Aufträge.
- Was machen Sie, wenn ein neuer Auftrag hereinkommt? Was machen Sie, wenn ein Auftrag erledigt ist und Arbeiter und Betriebsmittel wieder frei werden?
- Wie würden Sie sicherstellen, dass kein Auftrag unendlich lang unerledigt bleibt?
- Nehmen Sie an, dass ihre Arbeiter nicht immer zuverlässig sind. Sie vergessen manchmal, das Werkzeug zurückzubringen oder Sie schlafen manchmal ein. Wie würden Sie die Abläufe abändern.

Python

Aufgabe 2 (*20 Punkte*) Wir wollen ein Programm schreiben, dass folgendes Problem löst. Es wurde in der Vorlesung unter Empfehlungssysteme eingeführt.
Sie bekommen eine Folge von Zahlen und Listen von Zahlen, etwa

$$[1, 3, 7], [2, 4, 1], [4, 2], 2, \ldots$$

Dabei entsprechen die Listen Bestellungen und die Zahlen dem Aufrufen einer Produktseite. Wir wollen zu jedem Produkt, dass sich der Benutzer ansieht (also jeder Zahl in der Eingabefolge) drei Produkte anzeigen, die ihm auch gefallen können.
Bestimmen Sie dazu kontinuierlich wie häufig Produkte zusammen gekauft werden und geben Sie die drei Produkte aus, die am häufigsten mit dem derzeit betrachteten Produkt gekauft werden.
Im Beispiel ist 4 das am häufigsten mit 2 zusammen gekaufte Produkt.

Hinweis: mit `isinstance(a,list)` können Sie in Python feststellen, ob die Variable `a` einen Wert vom Typen Liste hat.