Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. S. Albers
T. Jacobs, F. Müller

Übungen zur Vorlesung Informatik II
Freiburg i. Br., den 8. Mai 2006

3. Übungsblatt

Aufgabe 1: Turm von Hanoi **10 Punkte**

Das Spiel *Tower of Hanoi* besteht aus einem Brett, auf dem nebeneinander 3 Stäbe befestigt sind, und einer Menge von n gelochten Scheiben unterschiedlichen Durchmessers, die an den Stäben gestapelt werden können. Zu Anfang sind alle Scheiben auf dem linken Stapel ihrer Größe nach von unten nach oben geordnet, wobei die größte Scheibe unten liegt. Ziel des Spiels ist es, dass sich alle Scheiben auf dem rechten Stapel befinden. In jedem Spielzug kann die jeweils oberste Scheibe von einem Stapel genommen und auf einen anderen Stapel gelegt werden. Jedoch darf zu keinem Zeitpunkt des Spiels eine größere Scheibe auf einer kleineren liegen.

a) Offensichtlich kann das Spiel folgendermaßen rekursiv gelöst werden: Ist $n = 1$, so bewege die einzige Scheibe vom linken auf den rechten Stapel. Gibt es n Scheiben, so bewege zunächst die ersten $n-1$ Scheiben genauso auf den mittleren Stapel, wie sie im Fall von $n-1$ Scheiben auf den rechten Stapel bewegt würden. Nimm nun die n-te Scheibe vom linken Stapel und lege sie auf den rechten. Anschließend bewege die $n-1$ Scheiben von der Mitte nach rechts, wieder so wie im Fall von $n-1$ Scheiben.
Stellen Sie eine Rekursionsformel für die Laufzeit $T(n)$ von diesem Algorithmus auf. Rechnen Sie die Laufzeit für einige kleinere Beispiele durch oder wenden Sie iteratives Einsetzen an, so dass Sie zu einer Vermutung für eine nicht rekursive Formulierung von $T(n)$ kommen. Beweisen Sie ihre Vermutung mittels vollständiger Induktion. **5 Punkte**

b) Implementieren Sie ein Java-Programm, welches das Spiel löst. Die Anzahl der Scheiben soll dabei als Kommandozeilen-Parameter übergeben werden. Für jeden Schritt, in dem eine Scheibe von einem Stapel auf einen anderen bewegt wird, soll das Programm genau folgende Textzeile ausgeben: „`Bewege Scheibe von` A `nach` B“, wobei $A, B \in \{\texttt{LINKS}, \texttt{MITTE}, \texttt{RECHTS}\}$ den entsprechenden Stapel repräsentiert.
Kommentieren Sie ihr Programm ausreichend, so dass ihr Tutor es verstehen kann. Für nicht kommentierte Programme werden keine Punkte vergeben. **5 Punkte**

Aufgabe 2: Minimale Puffergröße **5 Punkte**

Der bekannte Hydrologe Harold Erwin Hurst arbeitete in der ersten Hälfte des 20 Jahrhunderts am Assuan-Staudamm in Ägypten. Er beschäftigte mit der Frage, wie man den Wasserfluss des Nil regulieren könne, um Dürreperioden und Überschwemmungen vorzubeugen. Er wollte wissen, wie groß ein Wasserreservoir sein muss, mit dessen Hilfe man die Wassermenge pro Jahr konstant halten kann, indem man in regenreichen Jahren gestautes Wasser in trockenen Jahren dem Nil zusätzlich zuführt.
Sei $X_1, \ldots, X_n$ die jeweilige Wassermenge in den aufeinanderfolgenden Jahren $1, \ldots, n$. Gesucht ist die Größe R eines Wasserreservoirs, so dass in jedem Jahr dem Nil die gleiche Wassermenge zugeführt werden kann und dass am Ende des n-ten Jahres der Wasserstand im Reservoir wieder so hoch ist wie vor dem ersten Jahr. Von Störfaktoren wie Verdunstung und Versandung soll abgesehen werden.
Geben Sie einen Algorithmus an, welcher R berechnet und dessen Laufzeit in $O(n)$ liegt.

Abgabe: Montag, 15. Mai, 14 Uhr, in den entsprechenden Briefkästen im Erdgeschoss von Gebäude 051.

Die Übungsblätter können in Gruppen à **maximal 2 Personen** bearbeitet werden. Vermerken Sie die Namen und Matrikelnummern der an der Bearbeitung beteiligten Personen.

Beachten Sie bitte auch die aktuellen Hinweise unter

`www.informatik.uni-freiburg.de/~ipr` ⟶ `Teaching` ⟶ `Informatik II`