## Übungsblatt 2 für "Diskrete und geometrische Algorithmen"

1.) Sei $A[1..n]$ ein Feld mit $n$ verschiedenen Zahlen. Wenn $i < j$ und $A[i] > A[j]$ gilt, dann wird das Paar $(i, j)$ eine Inversion von $A$ genannt.

   (a) Welches Feld mit Elementen der Menge $\{1, 2 \ldots, n\}$ besitzt die meisten Inversionen und wie viele Inversionen sind in diesem Feld enthalten?

   (b) Wie ist die Beziehung zwischen der Laufzeit von Insertion Sort und der Anzahl der Inversionen im Eingabefeld? (Begründete Anwort!)

   (c) Geben Sie einen Algorithmus an, der die Anzahl der Inversionen einer Permutation von $n$ Elementen bestimmt und dessen Laufzeit im schlechtesten Fall $\Theta(n \log n)$ ist.
   **Hinweis:** Modifizeren Sie Merge Sort in geeigneter Weise.

2.) Die Fibonacci-Zahlen seien wie in der Vorlesung durch die Rekursion $F_n = F_{n-1} + F_{n-2}$, für $n \geq 2$, mit den Anfangswerten $F_0 = 0$ und $F_1 = 1$ definiert.

   (a) Betrachten wir den folgenden einfachen rekursiven Algorithmus zur Berechnung der Zahlen $F_n$:

```
FIB(n)
IF n = 0 THEN
   RETURN 0
ELSIF n = 1 THEN
   RETURN 1
ELSE
   RETURN FIB(n − 1) + FIB(n − 2)
END IF
```

   Offenbar hat man hierbei das Problem, daß in der Rekursion niedrigere Fibonacci-Zahlen oft neu bererechnet werden müssen. Überlegen Sie sich beispielsweise, wie oft zur Berechnung von $F_n$ die Zahl $F_2$ berechnet werden muß. Folgern Sie daraus, daß allgemein die Schrittzahl zur rekursiven Berechnung von $F_n$ exponentiell wächst.

   (b) Etwas genauer: stellen Sie eine Rekusion für die Anzahl der Additionen $A(n)$, die im obigen Algorithmus bei der Berechnung von $F_n$ gemacht werden auf und lösen Sie diese.

3.) Die Fibonacci-Zahlen können effizient mit Hilfe der folgenden auf Matrizenmultiplikation beruhenden Formel berechnet werden:

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n, \quad \text{für } n \geq 1.$$

(a) Verifizieren Sie diese Formel durch vollständige Induktion.

(b) Überlegen Sie sich einen rekursiven Algorithmus, der zur Berechnung von $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n$ nur logarithmisch viele Schritte benötigt.

4.) (a) Durch Aufstellen einer Rekursion und lösen derselben bestimmen Sie die Anzahl $M_n$ der Teilmengen der Menge $[n] := \{1, 2, \ldots, n\}$, die keine zwei aufeinanderfolgenden Zahlen enthalten.

(b) Bestimmen Sie weiters die Anzahl $C_n$ der Teilmengen von $[n]$, welche keine zwei aufeinanderfolgenden Zahlen enthalten, wobei jetzt zusätzlich 1 als Nachfolger von $n$ gilt (zyklische Anordnung).

5.) Bestimmen Sie die allgemeine Lösung der Rekursion

$$a_n = \tfrac{n+2}{3n} a_{n-1} + n^2 + 3n + 2.$$

6.) Lösen Sie das System von Rekursionen

$$a_{n+1} = 2a_n + 4b_n, \qquad b_{n+1} = 3a_n + 3b_n, \quad n \geq 0,$$

mit den Anfangswerten $a_0 = 1$ und $b_0 = -1$ indem Sie das System in eine äquivalente Rekursion zweiter Ordnung umformen.

7.) Nochmals zu den Fibonacci-Zahlen $F_n$: sei $(G_n)_{n\geq 0}$ die Folge der Quadrate der Fibonacci-Zahlen, also $G_n = F_n^2$. Man überlege sich, daß $G_n$ ebenfalls eine homogene lineare Rekursion mit konstanten Koeffizienten erfüllt und bestimme solch eine.

8.) Die Türme von Hanoi: Gegeben seien 3 Stäbe $A$, $B$, $C$ und $n$ verschiedene Scheiben. Anfangs seien alle Scheiben auf Stab $A$ der Größe nach aufgereiht, die größte ganz unten. Dieser Turm von Scheiben soll nun von $A$ nach $B$ transferiert werden, unter der Bedingung, daß in jedem Zug nur eine Scheibe bewegt und niemals eine größere Scheibe über einer kleineren platziert werden darf. Seien $a_n$ die minimale Anzahl der benötigten Züge und $b_n$ die minimale Anzahl benötigter Züge, wenn Bewegungen zwischen $A$ und $B$ nicht erlaubt sind. Bestimmen Sie $a_n$ und $b_n$ durch Aufstellen und Lösen einer Rekursion.