> *Im Großen Tempel von Benares, unter dem Dom, der die Mitte der Welt markiert, ruht eine Messingplatte, in der drei Diamantnadeln befestigt sind, jede eine Elle hoch und so stark wie der Körper einer Biene. Bei der Erschaffung der Welt hat Gott vierundsechzig Scheiben aus purem Gold auf eine der Nadeln gesteckt, wobei die größte Scheibe auf der Messingplatte ruht und die übrigen, immer kleiner werdend, eine auf der anderen. Das ist der* Turm von Brahma. *Tag und Nacht sind die Priester unablässig damit beschäftigt, den festgeschriebenen und unveränderlichen Gesetzen von Brahma folgend, die Scheiben von einer Diamantnadel auf eine andere zu setzen, wobei der oberste Priester nur jeweils eine Scheibe auf einmal umsetzen darf und zwar so, dass sich nie eine kleinere Scheibe unter einer größeren befindet. Sobald dereinst alle vierundsechzig Scheiben von der Nadel, auf die Gott sie bei der Erschaffung der Welt gesetzt hat, auf eine der anderen Nadeln gebracht sein werden, werden der Turm samt dem Tempel und allen Brahmanen zu Staub zerfallen, und die Welt wird mit einem Donnerschlag untergehen.*

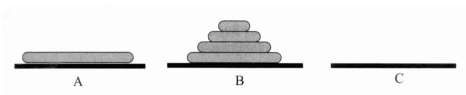
Im Jahre 1883 erfand der französische Mathematiker EDOUARD LUCAS[1] diese Geschichte, die heute meist als die Legende der *Türme von Hanoi* gilt. Auf der Pariser Weltausstellung stellte LUCAS seine Türme als mathematisches Spiel unter dem Pseudonym CLAUS[2] aus . Da ein Turm entfernt an eine Pagode erinnert, kam wahrscheinlich später der Name *Türme von Hanoi* auf. Als Spielzeug kann man es auch heute noch kaufen.

[1] FRANÇOIS EDOUARD ANATOLE LUCAS wurde am 4.4.1842 geboren und starb am 3.11.1891. Er war Professor für Mathematik am Lycée Saint Louis in Paris und am Lycée Charlemagne, also auch in Paris. Er hat sich sehr viel mit Zahlentheorie beschäftigt. Er fand die explizite Darstellung der Fibonacci-Zahlen.

[2] was durch Buchstabenvertauschung aus Lucas entstand.

Problemstellung: Ein Turm der Höhe n soll von der Stelle A nach Stelle C (mit Ablagestelle B) so transportiert werden, dass man immer nur eine Scheibe nehmen kann und niemals eine größere Scheibe über einer kleineren liegt.

Problemlösung: Die entscheidende Stelle bei der Umschichtung ist hier dargestellt und liefert die Idee für den Algorithmus:



Zur Erinnerung: „Rekursion heißt, ein Problem der ‚Schwierigkeitsstufe' n auf ein Problem der ‚Schwierigkeitsstufe' $n-1$ zurück zu führen."

Aufgabe: Formuliere die Idee zum Algorithmus in deutscher Sprache!

Die Lösung mit Java:

```java
 class Hanoi
{ int starthoehe;
  int bewegungen = 0;
  SimpleInput in;

  void bewege(char von, char nach)
  { Out.println("Scheibe von " + von + " nach " + nach);
    bewegungen++;
  }

  void turm(char start, char ziel, char ablage, int hoehe)
  { if (hoehe>1) turm(start, ablage, ziel, hoehe-1);
    bewege(start, ziel);
    if (hoehe>1) turm(ablage, ziel, start, hoehe-1);
  }

  public void action()
  {  in = new SimpleInput();
     starthoehe = in.getInt("Wie hoch soll der \"Turm von Hanoi\" sein? ");
     Out.println("Die Ausgabe fuer einen \"Turm von Hanoi\"" +
                 " der Hoehe " + starthoehe + ".");
     Out.println("Die Staebe sind mit A, B und C bezeichnet.");
     Out.println("Der Turm soll von A nach C transportiert werden.");
     turm('A', 'C', 'B', starthoehe); // von A nach C ueber B
     Out.println("Es waren " + bewegungen +
                 " Bewegungen fuer die Turmhoehe "
                 + starthoehe + " noetig.");
  }
} // class Hanoi
```

Der Aufrufmechanismus der Rekursion soll hier am Beispiel eines Turms der Höhe 3 dargestellt werden. Dabei sind die jeweils aktuellen Parameterwerte schon eingetragen (Anführungszeichen sind weggelassen worden).

```
void turm(A,C,B,3){
if (3 > 1) turm(A,B,C,2); ⟶  void turm(A,B,C,2){
                              if (2 > 1) turm(A,C,B,1); ⟶  void turm(A,C,B,1){
                                                            if (1 > 1) ... //Ende
                                                            bewege(A,C);
                                                            if (1 > 1) ... //Ende
                                                            }
                              bewege(A,B);
                              if (2 > 1) turm(C,B,A,1); ⟶  void turm(C,B,A,1){
                                                            if (1 > 1) ... //Ende
                                                            bewege(C,B);
                                                            if (1 > 1) ... //Ende
                                                            }
                              }
bewege(A,C);
if (3 > 1) turm(B,C,A,2); ⟶  void turm(B,C,A,2){
                              if (2 > 1) turm(B,A,C,1); ⟶  void turm(B,A,C,1){
                                                            if (1 > 1) ... //Ende
                                                            bewege(B,A);
                                                            if (1 > 1) ... //Ende
                                                            }
                              bewege(B,C);
                              if (2 > 1) turm(A,C,B,1); ⟶  void turm(A,C,B,1){
                                                            if (1 > 1) ... //Ende
                                                            bewege(A,C);
                                                            if (1 > 1) ... //Ende
                                                            }
                              }
} // Ende von turm(A,C,B,3)
```

Aufgaben:

1. Die Anzahl $a(n)$ der nötigen Scheibenbewegungen lässt sich durch eine rekursive Vorschrift formulieren. Finde diese Vorschrift und schreibe eine rekursive Funktion `int a(int n)`, die die Anzahl der Scheibenbewegungen ermittelt.

2. Schaut man sich die Folge $a(n)$ genauer an, dann erkennt man, dass es eine Formel zur direkten, nichtrekursiven Berechnung von $a(n)$ gibt. Eine Formel zur direkten Berechnung heißt auch *explizite* Formel.

   Freiwillige Zusatzaufgabe: Beweise die explizite Formel zur Berechnung von $a(n)$.

3. Wie lange dauert es laut der eingangs erwähnten Geschichte, bis die Welt untergeht, wenn ein Priester für eine Scheibe eine Sekunde benötigt?

4. Heutige Computer takten im GHz-Bereich. Wie lange dauert es bis zum Weltuntergang, wenn pro Takt eine Scheibe umgelegt wird?

Ergänzungen:

1. Buneman und Levy haben 1980 einen Lösungsalgorithmus beschrieben, der ebenfalls funktioniert und ohne Rekursion auskommt:

   Wiederhole, bis der gesamte Turm verschoben wurde:

   a) Die kleinste verschiebbare Scheibe auf den Stapel rechts von ihr legen (bzw. auf den ersten, wenn die Scheibe auf dem letzten Stapel liegt).

   b) Die zweitkleinste verschiebbare Scheibe auf den einzig möglichen Stapel verschieben.

2. Das entsprechende Problem für vier und mehr Stäbe bzw. Plätze ist bis heute noch nicht in allgemeiner Form gelöst worden.

3. Auf `http://www.kernelthread.com/hanoi/` findet man das Problem in 110 verschiedenen Programmiersprachen gelöst, dabei auch ein Betriebssystem für Intel-Prozessoren, das nichts anderes tut als das Turmproblem zu lösen.

4. Auf `http://www.gierhardt.de/informatik/bluej/Rekursion/index.html` findet man eine DOS-Datei `HANOI.EXE` (mit Modula-2 ca. 1992 geschrieben) eines verrückten Informatikers, die das Umsetzen der Scheiben demonstriert, wobei die Scheiben klingen und nach der pentatonischen Blues-Tonleiter gestimmt sind:

   „The Hanoi Blues Machine!“

   Das Programm sollte im Vollbildmodus laufen (Alt-Return nach Start im Fenster)[3].

   Viel Spaß damit!

[3] Unter Windows-7, -8 und -10 läuft das Programm nicht mehr direkt, sondern z.B. in der DOSBox.