Prof. Dr. Bernhard Zimmermann
FB Automatisierung und Informatik
Professor für Sprachen und Compilerbau

▲ Hochschule Harz
Hochschule für angewandte Wissenschaften

LV „Programmierung 1“

Praktikumsaufgabe 3

Nim-Spiel

Problembeschreibung

Das NIM-Spiel ist ein Spiel, bei dem 2 Spieler gegeneinander antreten. Es hat die besondere Eigenschaft, dass jede Spielerin schon aus dem Anfangsspielzustand feststellen kann, ob sie oder ihr Gegner das Spiel gewinnen wird, vorausgesetzt keiner der beiden macht einen Fehler.

Spielregeln:

Das Spiel beginnt mit drei Reihen, die beliebig viele Streichhölzer enthalten, z. B.

```
1. Reihe: |||
2. Reihe: ||||
3. Reihe: |||||  ||||
```

Zwei Spieler ziehen abwechselnd. Ein Zug besteht darin, eine Reihe auszuwählen und aus dieser Reihe beliebig viele, jedoch mindestens ein Streichholz zu entnehmen. Wer das letzte Streichholz nimmt, hat das Spiel gewonnen.

```
do {
```

Spielstrategie

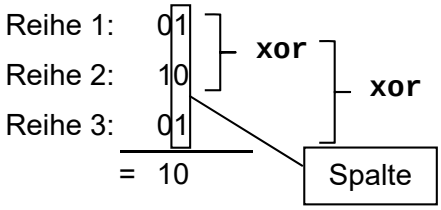
Die Strategie im NIM-Spiel besteht darin, dem Gegner in den 3 Reihen die als binäre Bitfolgen dargestellten Anzahlen n_1, n_2 und n_3 von Streichhölzern so zu hinterlassen, dass sich

$$n_1 \texttt{ xor } n_2 \texttt{ xor } n_3 = 0$$

ergibt, wobei **`xor`** die Exklusiv-Oder-Operation darstellt. Das Ergebnis 0 dieser Bit-Operation besagt, dass der Gegner mit seinem nächsten Zug auf keinen Fall in der Lage sein wird, das Spiel zu beenden und somit zu gewinnen, da nur das Vorhandensein von mindestens zwei Streichholzreihen mit sich gegenseitig aufhebenden Bitfolgen oder drei leere Reihen dieses Ergebnis liefern kann, wobei im letzteren Fall das Spiel gewonnen ist. Die Operation **`xor`** stellt praktisch einen bit-bezogenen, positionsweisen Test über alle drei Reihen auf gerade oder ungerade Bitsummen dar, d.h. die **`xor`**-Operation liefert für eine Bitposition genau dann 0, wenn die Summe der Bit-Werte, die in den drei Reihen an dieser Position auftreten, also quasi in einer Spalte, gerade ist. Ein Ergebnis größer als 0 würde demzufolge bedeuten, dass die Anzahl nichtleerer Reihen ungerade ist oder (!) sich die Reihen in mindestens einer Bitposition unterscheiden, und somit der Gegner die Möglichkeit hätte, das Spiel erfolgreich zu beenden.

Beispiel:

Reihe 2 enthält zwei Streichhölzer und die beiden anderen Reihen je ein Streichholz.



, d.h. an den Bitposition 2^0 ist die Spaltensumme über alle drei Reihen gerade, für die Bitposition 2^1 dagegen ungerade. Nach dem Entfernen von 2 Streichhölzern aus der zweiten Reihe werden sowohl die Spaltensummen als auch die Anzahl der gefüllten Reihen gerade und es besteht für den Gegner keine Gewinnmöglichkeit mehr.

Im folgenden bezeichnet n_i, i=1,2,3, die Anzahl der Streichhölzer in der Reihe i. Wenn eine Spielerin ihrem Gegner einen Spielzustand zum Ziehen übergibt, in dem

n_1 **xor** n_2 **xor** $n_3 = 0$

gilt, wird aufgrund der Spielregeln nach dem gegnerischen Zug immer

n_1 **xor** n_2 **xor** n_3 != 0

gelten, und die Spielerin kann mit ihrer Antwort wieder

n_1 **xor** n_2 **xor** $n_3 = 0$

erreichen, was aufgrund der obigen Betrachtungen schließlich zum Sieg führt.

Eine Schwierigkeit besteht lediglich noch darin zu bestimmen, in welcher Reihe wie viele Streichhölzer zu entfernen sind, wobei auch hier wieder das Ergebnis E des Ausdrucks

n_1 **xor** n_2 **xor** n_3

verwendet werden kann, da dieses die Spalten mit ungeraden Bitsummen liefert. Wird E mittels **xor** auf die Streichholzanzahl n_i einer Reihe angewendet, entsteht eine Bitfolge B, bei der in den durch die '1'-Bitstellen von E bestimmten Spalten die inversen bzw. negierten Bits aus den korrespondierenden Spalten von n_i auftreten und an den '0'-Bitstellen die unveränderten Bits aus den entsprechenden Spalten von n_i stehen. Die Bitfolge B legt also fest, wie die Anzahl der Streichhölzer für diese Reihe aussehen müsste, um die Gesamtbedingung n_1 **xor** n_2 **xor** $n_3 = 0$ zu erfüllen, weil sich für eine Spalte mit ungerader Bitsumme durch Umdrehen genau eines Bits eine gerade Bitsumme ergibt.

Das Ergebnis B kann kleiner oder größer als die ursprüngliche Streichholzanzahl n_i sein. Da die Ausgangssumme in den modifizierten Spalten ungerade war, enthält jede dieser Spalten mindestens eine 1. Durch Umdrehen eines solches Bits in der größten modifizierten Spalte wird die Anzahl der Streichhölzer in der entsprechenden Reihe kleiner, so dass sich bei der Anwendung dieser Berechnungsvorschrift auf alle Reihen mindestens eine Anzahl ergibt, die kleiner als die ursprüngliche ist. Da die Spielregeln es nur gestatten Streichhölzer aus den Reihen zu entfernen, kann dieser Wert direkt für den nächsten Zug als neue Anzahl von Streichhölzern in der entsprechenden Reihe verwendet werden.

Beispiel:

	Anzahl	Binärdarstellung
Reihe 1	3	011
Reihe 2	4	100
Reihe 3	5	101

Am Zug ist die erste Spielerin.

Nach der Berechnung von

011 **xor** 100 **xor** 101 = 010

wird jede Anzahl mit diesem Ergebnis mittels **xor** verknüpft, d.h. die jeweilige Bitposition 2^1 wird gedreht.

	Anzahl	Binärdarstellung	Anzahl xor 010
Reihe 1	3	011	001
Reihe 2	4	100	110
Reihe 3	5	101	111

Das Ergebnis der Reihe 1 ist kleiner als die Anzahl der Streichhölzer in Reihe 1 und ersetzt somit die aktuelle Streichholzanzahl dieser Reihe. Dieser Zug hinterlässt also in den Reihen die folgenden Streichholzanzahlen:

	Anzahl	Binärdarstellung
Reihe 1	1	001
Reihe 2	4	100
Reihe 3	5	101

Falls die erste Spielerin keinen Fehler mehr macht, ist sie nicht mehr am Gewinnen zu hindern.

```
} while (!verstanden);
```

Aufgabenstellung

Implementieren Sie das NIM-Spiel in Java, ***wobei nur die bis einschließlich Seite 99 des Vorlesungsskripts Programmierung-1 vorgestellten Sprachkonstrukte verwendet werden dürfen***. Ihr Programm soll zunächst den Benutzer über die Konsole auffordern, für jede Reihe die Startanzahl der Streichhölzer einzugeben. Anschließend kann der Benutzer dann gegen den Computer NIM spielen. Für jeden seiner Züge muss der Benutzer nach Aufforderung die Nummer der Reihe und die Anzahl der Streichhölzer, die aus dieser Reihe entfernt werden sollen, eingeben. Ihr Programm muss ungültige Benutzereingaben erkennen, z. B. eine falsche Reihe oder eine negative bzw. zu große Anzahl der aus einer Reihe zu entfernenden Streichhölzer. ***Der Computer muss nach der oben dargestellten Strategie spielen*** und nach jedem Zug den aktuellen Spielstand, wie in dem nachfolgenden Protokoll dargestellt, auf der Konsole ausgeben.

```
java Nim


NIM-Spiel
---------
Streichhoelzeranzahl der 1. Reihe: 3
Streichhoelzeranzahl der 2. Reihe: 4
Streichhoelzeranzahl der 3. Reihe: 9

Aktueller Spielstand:
---------------------
1. Reihe: |||
2. Reihe: ||||
3. Reihe: |||||  ||||

Sie sind am Zug!
Aus welcher Reihe moechten Sie Streichhoelzer entnehmen? 3
Wieviele Streichhoelzer moechten Sie entnehmen? 2

Aktueller Spielstand:
---------------------
1. Reihe: |||
2. Reihe: ||||
3. Reihe: |||||  ||

Der Computer ist am Zug!

Aktueller Spielstand:
---------------------
1. Reihe: ||
2. Reihe: ||||
3. Reihe: |||||  ||

Sie sind am Zug!
Aus welcher Reihe moechten Sie Streichhoelzer entnehmen? 3
Wieviele Streichhoelzer moechten Sie entnehmen? 1

Aktueller Spielstand:
---------------------
1. Reihe: ||
2. Reihe: ||||
3. Reihe: |||||  |

Der Computer ist am Zug!

Aktueller Spielstand:
---------------------
1. Reihe: |
2. Reihe: ||||
3. Reihe: |||||  |
```

```
Sie sind am Zug!
Aus welcher Reihe moechten Sie Streichhoelzer entnehmen? 1
Wieviele Streichhoelzer moechten Sie entnehmen? 1

Aktueller Spielstand:
---------------------
1. Reihe:
2. Reihe: ||||
3. Reihe: |||||  |

Der Computer ist am Zug!

Aktueller Spielstand:
---------------------
1. Reihe:
2. Reihe: ||||
3. Reihe: ||||

Sie sind am Zug!
Aus welcher Reihe moechten Sie Streichhoelzer entnehmen? 2
Wieviele Streichhoelzer moechten Sie entnehmen? 1

Aktueller Spielstand:
---------------------
1. Reihe:
2. Reihe: |||
3. Reihe: ||||

Der Computer ist am Zug!

Aktueller Spielstand:
---------------------
1. Reihe:
2. Reihe: |||
3. Reihe: |||

Sie sind am Zug!
Aus welcher Reihe moechten Sie Streichhoelzer entnehmen? 3
Wieviele Streichhoelzer moechten Sie entnehmen? 2

Aktueller Spielstand:
---------------------
1. Reihe:
2. Reihe: |||
3. Reihe: |

Der Computer ist am Zug!

Aktueller Spielstand:
---------------------
1. Reihe:
2. Reihe: |
3. Reihe: |
```

☹ Fehler!

Der Computer wird nun gewinnen!

```
Sie sind am Zug!
Aus welcher Reihe moechten Sie Streichhoelzer entnehmen? 3
Wieviele Streichhoelzer moechten Sie entnehmen? 1

Aktueller Spielstand:
---------------------
1. Reihe:
2. Reihe: |
3. Reihe:

Der Computer ist am Zug!

Der Computer hat dieses Spiel gewonnen!

Process Nim finished
```

Benutzen Sie für die Entwicklung Ihres Programms Struktogramme.

Abzugeben ist neben den Struktogrammen ein ausführlich kommentierter Programmausdruck.

Spätester Abgabetermin:

Die Lösungen müssen der zuständigen Laborbetreuer*in nach der Abnahme spätestens während des **2**-ten zu dieser Aufgabe stattfindenden Labortermins per E-Mail übergeben werden.

Prof. Dr. Bernhard Zimmermann