

## Übungen zum Programmierkurs I

Abgabe bis zum Freitag, 09.05.2003, 20:00 Uhr

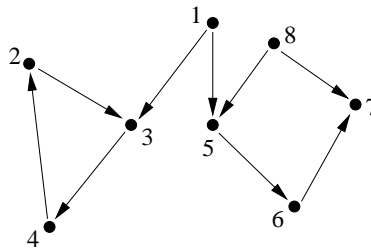
**Nur für Zweitsemester Wirtschaftsinformatik!**

### Aufgabe 14.1      **Kreise finden (mittel)**

**10 Punkte**

Gegeben sei ein *gerichteter* Graph, dessen Knoten mit 1 bis  $n$  bezeichnet sind (siehe auch Abschnitt 1.3.3.4 der Info I-Vorlesung). Man sagt, dass es einen *Weg* von Knoten  $i$  zu Knoten  $j$  gibt, wenn es eine Folge von Kanten gibt, die von  $i$  nach  $j$  führt. Ein *Kreis* sei ein Weg, der aus mindestens einer Kante besteht, und der von irgendeinem Knoten  $i$  wieder zu  $i$  führt.

Beispiel: Im unteren Graph bilden etwa die Knoten 2, 3 und 4 einen Kreis, nicht jedoch die Knoten 5 bis 8!



Schreiben Sie ein Programm, welches zu einem gegebenen gerichteten Graphen feststellt, ob es von Knoten 1 aus einen Weg zu einem Knoten gibt, der in einem Kreis liegt. Ist dies der Fall, sollen sowohl der betreffende Weg als auch der Kreis ausgegeben werden.

Die Graphen sollten aus Dateien eingelesen werden; auf der Kurs-Webseite finden Sie einige Beispieldateien. Diese enthalten in der ersten Zeile die Anzahl der Knoten. Alle anderen Zeilen sind von der Form  $i j$  und bedeuten, dass es eine Kante von Knoten  $i$  zu Knoten  $j$  gibt.

### Aufgabe 14.2      **Das Nim-Spiel (mittel)**

**10 Punkte**

Dieses Spiel läuft nach den folgenden Regeln:

Zu Beginn wird eine Anzahl Reiskörner in irgendeiner Weise auf verschiedene Haufen verteilt. Wie viele Haufen dabei gebildet werden, ist beliebig; auch kann die Anzahl der Reiskörner von Haufen zu Haufen verschieden sein. Zwei Spieler ziehen abwechselnd; ein Zug besteht daraus, von genau einem Haufen beliebig viele Reiskörner, mindestens jedoch eins, zu entfernen. Gewonnen hat, wer das letzte Reiskorn nimmt.

Implementieren Sie dieses Spiel. Der Hauptteil Ihres Programms sollte die Haufen aufbauen, die Spielzüge der beiden Spieler abfragen, für die Bildschirmausgabe sorgen, feststellen, ob jemand gewonnen hat etc.

Entwerfen Sie dazu einen Datentyp, der einen Spieler darstellt. Überlegen Sie, welche Elemente ein solcher Datentyp haben sollte – wichtig wäre z.B. eine Funktion, die den nächsten Spielzug bestimmt. Das Hauptprogramm könnte dann zwei Variablen vom Typ ‘Spieler’ verwalten und abwechselnd deren Funktionen aufrufen, um die Spielzüge abzufragen. Grenzen Sie das Hauptprogramm und die Spieler sinnvoll voneinander ab. So sollte der Spieler beispielsweise nicht in der Lage sein, regelwidrig zu ziehen.

Ihr Programm sollte zwei Arten von Spielern unterstützen: menschliche Spieler, die ihre Spielzüge etwa über die Tastatur eingeben, und ‘dumme’ computergesteuerte Spieler, die zufällig ziehen. Im Hauptprogramm soll man auswählen können, ob die beiden Spieler jeweils Menschen oder Computer sind.

### Aufgabe 14.3      **Das Nim-Spiel II (Zusatzaufgabe)**

**7 Punkte**

Erweitern Sie das Spiel um einen ‘klugen’ computergesteuerten Spieler.

Das Nim-Spiel hat die Eigenschaft, dass in jeder Spielsituation folgendes gilt:

- Entweder kann man so ziehen, dass man unabhängig von der Reaktion des Gegenspielers irgendwann zum Sieg kommt;
- oder es ist so, dass unabhängig von den eigenen Spielzügen der Gegner gewinnen kann.

Eine genaue Erläuterung, wie dies funktioniert, finden Sie z.B. auf der folgenden Webseite: <http://www.csm.astate.edu/Nim.html>. Implementieren Sie die dort beschriebene Strategie.

### Allgemeine Hinweise

- Dieses Übungsblatt ist **nur von den Zweitsemestern Wirtschaftsinformatik** zu bearbeiten!
- Die Webseite zum Programmierkurs ist wie folgt:  
<http://www.fmi.uni-stuttgart.de/szs/lehre/ss2003/ada95>
- Falls Sie Fragen irgendwelcher Art haben, wenden Sie sich bitte an Ihren Tutor oder an die Übungsleitung:  
[Stefan.Schwoon@fmi.uni-stuttgart.de](mailto:Stefan.Schwoon@fmi.uni-stuttgart.de), Raum 1.342, oder Tel. 7816-427.