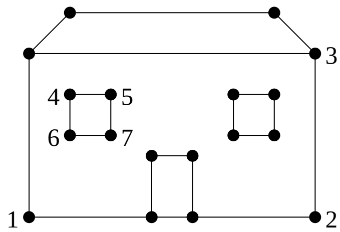
Ada-Kompaktkurs (Wirtschaftsinformatiker), Übungsblatt 13
Claus/N.Weicker, WS 02/03

Aufgabe 1: Komponenten (mittel) 10 Punkte

Ein ungerichteter Graph besteht aus einer Menge von Knoten und einer Menge von Kanten, die jeweils zwei Knoten miteinander verbinden (siehe auch Abschnitt 1.3.3.4 der Info I-Vorlesung). Man sagt, dass es einen *Weg* zwischen Knoten i und j gibt, wenn es eine Folge von Kanten gibt, die von i nach j (bzw. umgekehrt) führt. Eine *Komponente* eines Graphen ist eine Menge von Knoten, die alle paarweise durch Wege miteinander verbunden sind. Eine Komponente ist *maximal*, wenn man ihr keine weiteren Knoten hinzufügen kann.
Beispiel: Im unteren Graph bilden etwa die Knoten 1, 2 und 3 eine Komponente, die jedoch nicht maximal ist; Knoten 4 bis 7 bilden eine maximale Komponente. Insgesamt hat der Graph drei maximale Komponenten.



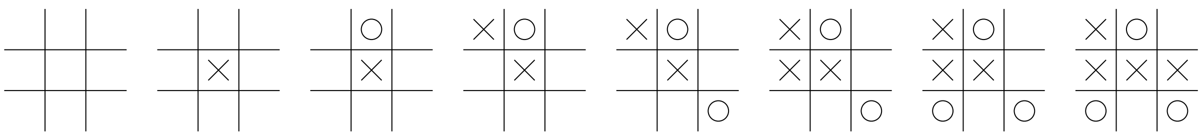
Schreiben Sie einen Programm, welches in einem gegebenen ungerichteten Graphen die maximalen Komponenten identifiziert. Die Graphen sollten aus Dateien eingelesen werden; auf der Kurs-Webseite finden Sie einige Beispieldateien. Diese enthalten in der ersten Zeile die Anzahl der Knoten, wobei wir im folgenden davon ausgehen, dass die Knoten Nummern von 1 bis n tragen. Alle anderen Zeilen sind von der Form $i\ \ j$, wobei $i < j$ gilt, und bedeuten, dass es eine Kante zwischen Knoten i und Knoten j gibt.

Aufgabe 2: Tic-Tac-Toe (mittel) 10 Punkte

Dieses Spiel kennen Sie sicher aus Ihrer Schulzeit (hoffentlich nicht aus den Vorlesungen). Falls nicht, erklären wir es hier nochmal:

> Das Spielfeld besteht aus 9 Feldern, die in drei Reihen und Spalten angeordnet sind. Anfangs sind alle Felder frei. Zwei Spieler ziehen abwechselnd; ein Zug besteht aus dem Besetzen eines beliebigen bis dahin freien Felds. Kann ein Spieler alle Felder einer Reihe, Spalte oder Diagonale besetzen, hat er sofort gewonnen. Das Spiel endet unentschieden, wenn das ganze Feld belegt wird, ohne dass ein Sieger feststünde.

Das Beispiel zeigt einen Spielablauf, der mit dem Sieg des ersten Spielers endet. Felder des ersten Spielers sind mit Kreuzen, die des zweiten mit Kreisen gekennzeichnet.
Implementieren Sie dieses Spiel. Der Hauptteil Ihres Programms sollte das Spielfeld verwalten, die Spielzüge der beiden Spieler abfragen, für die Bildschirmausgabe sorgen, feststellen, ob jemand gewonnnen 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 oder das Spielbrett auf unvorgesehene Weise zu manipulieren.



Ihr Programm sollte zwei Arten von Spielern unterstützen: menschliche Spieler, die ihre Spielzüge etwa über die Tastatur eingeben, und 'dumme' computergesteuerte Spieler, die bei jedem Zug zufällig ein freies Feld besetzen. Im Hauptprogramm soll man auswählen können, ob die beiden Spieler jeweils Menschen oder Computer sind.

Aufgabe 3: Tic-Tac-Toe II (Zusatzaufgabe) 7 Punkte

Erweitern Sie das Spiel um einen 'klugen' computergesteuerten Spieler.
Bekanntermaßen kann ein Tic-Tac-Toe-Spiel nicht gewonnen werden, wenn beide Spieler optimal agieren, d.h. "keine Fehler machen". Programmieren Sie Ihren Computer-Spieler mit einer optimalen Strategie, d.h. er sollte

- niemals verlieren;
- stets gewinnen, wenn der Gegner es erlaubt.

Hinweise

- Falls Sie Fragen irgendwelcher Art haben, wenden Sie sich bitte an Ihren Tutor oder an die Übungsleitung: `Nicole.Weicker@informatik.uni-stuttgart.de` oder Tel. 7816-412