Übungen zur Vorlesung
Grundlagen der Programmierung II
Blatt 7

Aufgabe 1: Kantenanzahl in ungerichteten Graphen

Zeigen Sie: Für jeden ungerichteten Graphen G=(V,E) gilt

$$\sum_{v \in V} d(v)=2|E|.$$

Aufgabe 2: Adjazenzlisten

a) Schreiben Sie eine Funktion in PASCAL, die einen Graph G=(V,E) mit V={1,...,n} einliest und als Ergebnis einen Zeiger auf die Adjazenzlistendarstellung des Graphen liefert. Die Eingabe bestehe aus der Zahl n gefolgt von den Kanten $\{i,j\} \in V$ in Form von Paaren i:j.

b) Schreiben Sie eine Prozedur zum Löschen eines Knoten aus dem Graphen.

c) Der **Abstand** zweier Knoten u, v ist definiert als die Länge des kürzesten Weges zwischen u und v. Schreiben Sie eine Funktion, die zu einem beliebigen Knoten v des Graphen eine Liste aller Knoten im Abstand 2 von v liefert.

Aufgabe 3: Dünnbesetzte Matrizen

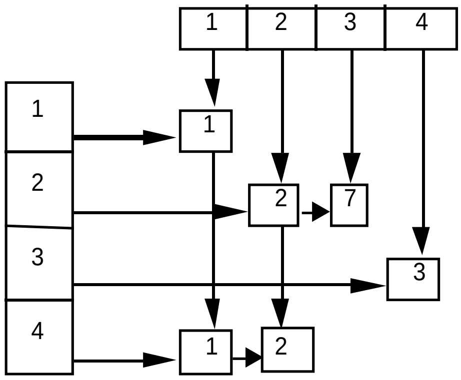
In der Praxis arbeitet man häufig mit sehr großen Matrizen (mehrere tausend Zeilen und Spalten), von denen aber meist viele Elemente den gleichen Wert besitzen (z.B. =0 sind), sog. **dünnbesetzte Matrizen**. Um Speicherplatz zu sparen, speichert man bei diesen Matrizen nur die relevanten Werte, also z.B. die $\neq 0$ ab. Dazu bedient man sich einer verketteten Struktur linearer Listen. Jede Zeile ist eine kreisförmig verkettete lineare Liste, in der die Elemente $\neq 0$ dieser Zeile aufgesammelt sind. Die einzelnen Zeilenlisten sind über ein Zeigerarray zugreifbar. Weiterhin ist jede Spalte eine kreisförmig verkettete Liste, in der die Elemente dieser Spalte $\neq 0$ eingetragen sind., ebenfalls zugreifbar über ein Zeigerarray. Jeder Matrixeintrag kommt aber nur einmal vor, er ist also sowohl in der Zeilenliste als auch in der zugehörigen Spaltenliste eingetragen.

Implementieren Sie dünnbesetzte Matrizen in PASCAL und entwerfen Sie eine Prozedur, die ein neues Element, gegeben durch Zeilen-, Spaltenindex und Wert in die Matrix einfügt.

Beispiel: Darstellung der Matrix

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 7 & 0 \\ 0 & 0 & 0 & 3 \\ 1 & 2 & 0 & 0 \end{matrix}$$

durch die oben beschriebene Struktur, wobei die kreisförmigen Verkettungen aus Gründen der Übersichtlichkeit weggelassen wurden:



Aufgabe 3: Rekursion und Iteration

Implementieren Sie folgende rekursive Funktionen nicht-rekursiv, indem Sie das Schema aus Abschnitt 3.8 verwenden:

a) Ackermann-Funktion f: IN→IN mit

$$f(m,n)= \begin{cases} n+m+1, \text{ falls } m*n=0, \\ \\ f(m-1,f(m,n-1)), \text{ falls } m*n\neq 0. \end{cases}$$

b) Die rekursive Funktion für das Mischen zweier sortierter Folgen (vgl. GdP I)

Testen Sie die rekursive und nicht-rekursive Version mit Hilfe ausgewählter Eingabewerte.

Kann man bei a) und b) auch ohne Stack auskommen?