Graphentheoret. 1. Übungsblatt (16.10.01)

1. (5 Punkte) Der Tiefensuchbaum, der bei der Tiefensuche erstellt wird, hängt im Allgemeinen von der Wahl des Ausgangsknotens und von der Ordnung der Adjazenzlisten ab.

Für welche **(a)** ungerichteten **(b)** gerichteten Graphen ist das Ergebnis unabhängig von der Ordnung der Adjazenzlisten? (bei festem Startknoten).

(c) Kann man durch geeignete Wahl des Startknotens und geeignete Ordnung der Adjazenzlisten jeden beliebigen Baum in einem ungerichteten Graphen zum Tiefensuchbaum machen?

2. (5 Punkte) Betrachten Sie das 3x4-Gitter, das aus den Knoten $\{(i,j) | 1 \leq i \leq 3,\ 1 \leq j \leq 4\}$ besteht, und in dem zwei Knoten benachbart sind, wenn sich ihre Koordinaten an genau einer Stelle unterscheiden, und zwar um ± 1. Die Knoten seien lexikographisch aufsteigend nummeriert ($k = 4(i-1)+j$). Nehmen wir an, dass die Adjazenzlisten nach der Nummer der Nachbarknoten aufsteigend sortiert sind.

(a) Berechnen Sie den Tiefensuchbaum und die TS-Nummerierung für den Startknoten $(2,2)$.

(b) Betrachten Sie die gerichtete Variante des Graphen, wo alle Kanten von links nach rechts $(i,j) \to (i,j+1)$ und von oben nach unten $(i,j) \to (i+1,j)$ gerichtet sind, und berechnen Sie den Tiefensuchbaum und die TS-Nummerierung für den Startknoten $(1,1)$.

3. Tiefensuche mit einem Stapel (5 Punkte)

Das folgende Programm ist eine nicht rekursive Variante für Tiefensuche, die mit einem Stapel arbeitet. Wir nehmen an, dass alle Knoten von v_1 aus erreichbar sind.

```
PUSH(v1)
While Stapel != nil do
  v:= POP()
  if TSnum(v)=0 then
    i:=i+1
    TSnum(v):=i
    fuer alle Kanten (v,w) do
      PUSH(w)
```

(a) Lösen Sie Aufgabe **2 (a)** für diesen Algorithmus. Was ist die Entsprechung zwischen diesem Algorithmus und dem in der Volesung vorgestellten rekursiven Algorithmus?

(b) Wie muss man den Algorithmus modifizieren, damit er auch funktioniert, wenn von v_1 nicht der ganze Graph erreichbar ist?