

# ARBEITSBLATT TIEFENSUCHE INFORMATIK GK 12 VOM 10.09.2007

In der letzten Stunde habt ihr den Algorithmus zur Tiefensuche kurz kennen gelernt. Eine grobe Formulierung dieses Algorithmus könnte wie folgt formuliert werden:

## Algorithmus Tiefensuche

Input: Startknoten: TGraphNode

- Markiere Startknoten
- Gehe an den Anfang der Kantenliste des Startknotens
- Für alle Kanten der gesamte Liste
  - tue: • Wenn Nachbarknoten nicht markiert ist
  - dann: • starte Tiefensuche mit Nachbarknoten

**Aufgabe 1:** Übersetze den Algorithmus in die Programmiersprache Delphi.

Hinweis: Die Schleife zum Durchlaufen aller Kanten der Kantenliste solltest du wieder mit einer `while not Node.edges.isBehind` – Schleife realisieren.

**Aufgabe 2:** Schreibe eine Testprozedur, welche anschließend anzeigt, ob der Graph zusammenhängend ist, oder nicht. (Hinweis: Zusammenhängend war gleichbedeutend damit, dass durch die Tiefensuche alle Knoten besucht wurden.)

**Aufgabe 3:** Erweitere die Prozedur Tiefensuche, so dass die Besuchsreihenfolge der Knoten ausgegeben wird. Zum Beispiel:

```
function TGraph.BesuchsreihenfolgeTiefensuche(startknoten: string): string;
var Reihenfolge: string;

    procedure Tiefensuche(Node: TGraphNode; var Reihenfolge: string);
    var Aktknoten: string; Zielknoten: string;
    begin
        ...
    end;

begin
    Reihenfolge:= '';
    Tiefensuche(searchNode(startknoten), Reihenfolge);
    result:= Reihenfolge;
end;
```

**Aufgabe 4:** Nutze nun diesen Algorithmus, um den kürzesten Weg von einem Startknoten zu einem Zielknoten zu finden.

```
function TGraph.kuerzesteDistanz(startknoten, zielknoten: string): double;
var Mindistanz: double;

    procedure Tiefensuche(AktNode, ZielNode: TGraphnode;
        Distanz: double; var MinDistanz: double);

    var ...
    begin
    end;

begin
    Mindistanz:= 1e100; // also "unendlich" groß
    Tiefensuche(searchNode(startknoten), searchNode(zielknoten), 0, Mindistanz);
    result:= Mindistanz;
end;
```