3 *Labormäuse in Labyrinthen* *Wichtung:* 20%

In der folgenden Abbildung ist ein bekanntes Labyrinth dargestellt. Erzeugen Sie den Graph, indem an jeder Stelle, wo mehr als ein Weg gewählt werden kann, ein Knoten angeordnet wird und die Knoten dann entsprechend den Wegen verbunden werden. Testen Sie die Tiefensuch- und Breitensuchmethode von Aufgabe 2 mit dem Graph. Der Graph soll aus einer Datei gelesen werden.

! *Zusätzliche Gewichte für Graphen*

Gewichtete Graphen sind Graphen, in denen mit jeder Kante Gewichte oder Kosten verknüpft sind. Das Problem des kürzesten Pfades besteht darin, in einen gewichteten Graph den zwei gegebene Knoten verbindenen Pfad zu finden, der die Eigenschaft hat, daß die Summe der Gewichte aller Kanten innerhalb der Menge aller solchen Pfaden minimal ist.
Definieren Sie eine Unterklasse von Graphen, die eine neue Methode für kürzesten Pfad enthält.

Hinweis: Um den kürzesten Pfad zwischen Knoten x und y zu finden wird ein Baum für den Knote x erzeugt, indem wir bei jedem Schritt den Knoten vom Rand hinzufügen, der x am nächsten liegt. Um zu ermitteln, welcher Randknoten x am nächsten liegt, verwenden wir ein Feld *val.* Für jeden Baumknoten k ist *val[k]* der kürzeste Abstand (welcher aus Baumknoten bestehen muß) von diesem Knoten zu x. Wenn k dem Baum hinzufügt wird, aktualisieren wir den Rand, indem wir die direkt mit k verbundene Knoten durchlaufen. Für jeden Knoten t in dem Rand wird der Wert *val[t]* neu berechnet. Dieser Prozeß wird fortgesetzt, bis entweder y dem Baum hinzufügt wird oder keine Knoten dem Baum hinzufügt werden können.

5. Übungsblatt

Ausgabe: Mo. 29.06.1998
Abgabe: Mo. 13.07.1998

---

1 *Hashing* *Wichtung:* 30%

Eine Hash-Funktion transformiert den Suchschlüssel in eine Tabellenadresse. Wenn ein Schlüssel (aus Zeichenfolge von Buchstaben) mit Hilfe des einfach Fünf-Bit-Codes codiert wird, wobei der *i*-te Buchstabe in Alphabet durch die Binärdarstellung der Zahl *i* dargestellt wird, so können wir den Schlüssel AKEY als die Binärzahl

00001010110010111001

ansehen, welche äquivalent zu der Dezimalzahl 44217 ist. Sei H die Größe der Hash-Tabelle, so ist der Hash-Code von AKEY *44217 mod H*.

Lineares Austesten ist eine einfache Methode zur Kollisionsbeseitigung: Wenn eine Kollision vorliegt, so teste man einfach die nächste Position in der Tabelle. Falls die Schlüssel übereinstimmen, ist die Suche erfolgreich beendet; falls auf dieser Position kein Datensatz vorhanden ist, ist die Suche erfolglos beendet; andernfalls teste man wiederum die nächste Position, usw.

Implementieren Sie eine Java-Klasse mit den folgenden Konstruktor und Methoden:

public LinearHash(int x)	// wobei x die Größe der Hash-Tabelle ist
public int HashFunction(String s)	// basiert auf der Fünf-Bit-Code-Codierung
public void put(String s)	// Schlüssel s wird in die Tabelle eingefügt
public void remove(String s)	// Schlüssel s wird aus der Tabelle entfernt
public String get(int x)	// suche einen Schlüssel mit Hash-Code x

Vergessen Sie die Ausnahmebehandlung und das Testen nicht. Unter welchen Bedingungen ist das lineare Verfahren günstiger als das hashing mit getrennter Verkettung.

2 *Graphen* *Wichtung:* 50%

Diese Aufgabe handelt sich um Graphen und Algorithmen für Graphen. Ein Graph ist eine Menge von Knoten und Kanten. Knotennamen können einfach ganze Zahlen sein, z.B. in einer elektrischen Schaltung. Für gewisse Anwendungen, wie eine Karte der Fluglinien, ist es nicht sinnvoll, ganze Zahlen für die Knotennamen zu nehmen. Sie könnten aber in ganze Zahlen umgewandelt werden, indem wir eine Hash-Tabelle (siehe Aufgabe 1) führen. Einfache Operationen für Graphen sind, z.B., Einfügen und Entfernen von Knoten und Kanten. Anspruchsvolle Algorithmen für Graphen sind, z.B., Tiefensuche, Breitensuche und Anzahl der zusammenhängenden Komponenten.

Entwerfen Sie eine Schnittstelle, die die Methoden für Einfügen und Entfernen von Knoten und Kanten, Tiefensuche, Breitensuche und Anzahl der zusammenhängenden Komponenten enthält. Implementieren Sie die Schnittstelle. Verwenden Sie die Adjazenzstruktur als Darstellung für Graphen. (Führen von Hash-Tabellen für Knoten ist nur zu empfehlen.)