Technische Universität München
Institut für Informatik
Prof. Dr. Angelika Steger

SS 2001
Aufgabenblatt 2

Praktikum Algorithmen-Entwurf

(Abgabetermin: Montag, den 14.05.2001, 14.^{00} Uhr)

Minimale Spannbäume

Aufgabe 1 Algorithmus von Kruskal(mit Animation)

Implementieren und animieren Sie den Algorithmus von Kruskal zur Berechnung eines minimalen Spannbaums in einem ungerichteten, zusammenhängenden Graphen G mit positiven, reellwertigen Kantengewichten. Es soll am Bildschirm sichtbar sein, in welcher Reihenfolge die Kanten von G bearbeitet werden und welche Kanten bereits in den Spannbaum eingefügt wurden. Für den Algorithmus von Kruskal steht Ihnen frei, entweder die Implementierung mittels Priority-Queue oder die Implementierung mit Sortierung der Kanten zu wählen.

Als Eingabe stehen die Graphen `mst1.gw` bis `mst4.gw` an den üblichen Orten zur Verfügung. Die Kantengewichte dieser Graphen sind ganze Zahlen, die im User-Label in String-Form mit abgespeichert sind. Um diese Werte zwecks bequemer Weiterverarbeitung in einem `edge_array` abzulegen, kann eine Schleife der folgenden Form verwendet werden:

```
#include <LEDA/stream.h>
...
edge_array<int> c(g);
forall_edges(e,g) {
    string s = gw.get_user_label(e);
    string_istream I(s);
    I>>c[e];
}
```

Aufgabe 2 Algorithmus von Prim (große Graphen)

Implementieren Sie den Algorithmus von Prim zur Berechnung eines minimalen Spannbaums in einem ungerichteten, zusammenhängenden Graphen G mit positiven, reellwertigen Kantengewichten. Es soll am Ende das Gewicht des minimalen Spannbaums in der Ausgabedatei ausgegeben werden.

Auf der Homepage zu unserem Praktikum finden Sie Testeingabe und –ausgabedateien (*public.in* und *public.out*).
Ein Rahmenprogramm zum Einlesen der Eingabe wurde bereits zusammen mit Blatt 1 auf der Praktikums–Webseite

http://wwwmayr.in.tum.de/lehre/2001SS/optprak/data.html

bereitsgestellt. Dort findet sich ebenfalls ein Programm zum Erzeugen beliebig grosser Eingabegraphen.
Bitte beachten Sie, dass nur Loesungen akzeptiert werden, die tatsächlich effizient sind. Dies bedeutet unter anderem, dass Ihre Lösung auch Eingaben mit 1000 Knoten noch in vernünftiger Zeit verarbeiten können muss.

Eingabeformat:
#Probleme
#Knoten #Kanten
#Nr. Startknoten #Nr. Endknoten #Gewicht
...
#Knoten #Kanten
#Nr. Startknoten #Nr. Endknoten #Gewicht
...

Beispiel:
2
4 6
0 1 2
0 2 1
0 3 2
1 2 1
1 3 2
2 3 1
3 3
0 1 2
0 2 3
1 2 4

Ausgabeformat:
für jedes Problem: Gewicht eines minimal spannenden Baumes

Ausgabe zu obigem Beispiel:
3
5