Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Chris Pinkau, Stefan Toman, Jeremias Weihmann

Sommersemester 2013
Aufgabenblatt 2
22.04.2013

Praktikum Diskrete Optimierung

Due date: Monday, 29th April 2013, 14:00

Aufgabe 1 (Minimum Spanning Trees: `kruskal`)

Implement and animate the algorithm of Kruskal for computing a minimum spanning tree of an undirected connected graph G having positive integral edge weights. It should be visible on the screen in which order the edges of G are processed, and which edges are already chosen as spanning tree edges. For the algorithm of Kruskal you are allowed to choose between an implementation using a priority queue or, alternatively, an implementation which sorts the edges at the beginning.

As inputs you can use the graphs `mst1.gw` to `mst4.gw` which are available at the website. The edge weights of these graphs are integers which are stored as strings at the user label. For the purpose of a more comfortable processing you can use a loop like the following to store these values in an `edge_array`:

```
#include <LEDA/system/stream.h>
...
leda::edge_array<int> c(g);
edge e;
forall_edges(e,g){
    leda::string s = gw.get_user_label(e);
    leda::string_istream I(s);
    I >> c[e];
    std::cout << c[e] << "\n";
}
```

Aufgabe 2 (Minimum Spanning Trees: `prim`)

Analogously to the first assignment, implement and animate the algorithm of Prim for computing a minimum spanning tree.