Prof. Dr. Jens Vygen
Pietro Saccardi, M.Sc.

Exercise Set 3

Exercise 3.1:

Definition. *For a finite set* $\emptyset \neq T \subset \mathbb{R}^2$ *we define*

$$\mathrm{BB}(T) := \max_{(x,y)\in T} x - \min_{(x,y)\in T} x + \max_{(x,y)\in T} y - \min_{(x,y)\in T} y.$$

Moreover we denote with $\mathrm{Steiner}(T)$ *the length of a shortest rectilinear Steiner tree for* T*, and with* $\mathrm{MST}(T)$ *the length of a minimum spanning tree in the complete graph on* T*, where the cost of an edge* (v, w) *is* $\ell_1(v, w)$*.*

Prove that:

(a) $\mathrm{BB}(T) \leq \mathrm{Steiner}(T) \leq \mathrm{MST}(T)$;

(b) $\mathrm{Steiner}(T) = \mathrm{BB}(T)$ for $|T| \leq 3$;

(c) $\mathrm{Steiner}(T) \leq \frac{3}{2}\,\mathrm{BB}(T)$ for $|T| \leq 5$;

(d) There is no $\alpha \in \mathbb{R}$ s.t. $\mathrm{Steiner}(T) \leq \alpha\,\mathrm{BB}(T)$ for all finite $\emptyset \neq T \subset \mathbb{R}^2$.

(2+2+3+2 points)

Exercise 3.2:
Consider a finite nonempty set of terminals $T \subseteq V(G) \subset \mathbb{R}^2$; assume the ℓ_1-distance as cost function for the edges. Show that the bounding box net length is a feasible lower bound ($\mathrm{lb}(v, I) := \mathrm{BB}(\{v\} \cup I)$).

(3 points)

Turn page! $\longrightarrow$

Exercise 3.3: Programming exercise

Implement the Dijkstra-Steiner algorithm for 3D grid graphs. Assume $c(v,w) = \ell_1(v,w)$. Your program will take an input in the form

$x_1 \ x_2 \ x_3 \ \cdots \ x_{n_x}$

$y_1 \ y_2 \ y_3 \ \cdots \ y_{n_y}$

$z_1 \ z_2 \ z_3 \ \cdots \ z_{n_z}$

$t_{1x} \ t_{1y} \ t_{1z} \ t_{2x} \ t_{2y} \ t_{1z} \ \cdots \ t_{kx} \ t_{ky} \ t_{kz}$

where all the numbers are space-separated nonnegative integers, and x_i, y_j, z_k describe the coordinates of the grid along the x-, y- and z-axis respectively, and $T_l := (t_{lx}, t_{ly}, t_{lz})$ are the coordinates of the l-th terminal.
Your program shall check for the input correctness, and return exit code 1 if a terminal does not lie on the grid. Otherwise, it must compute the length of a shortest Steiner tree, and print this value to the standard output. The program should achieve the claimed runtime; you do not have to implement a Fibonacci heap, you may use any other heap structure available in the standard library. You should expect test instances of up to 20 terminals and approximately 10 000 vertices.

The program must be written in C or C++ (you are allowed to use up to C++11) and must compile and run on Linux. To achieve the maximum score, your program must not leak any memory. It must compile with either Clang $\geq$ 3.4.2 or Gcc $\geq$ 4.8.3, with `-Wall -Wpedantic -Werror`, and it cannot link to any other library. You can use any tool available in the standard library.

The deadline for the programming exercises is May 22$^{\text{nd}}$, 12:00. Deliver your source code by email.

(15 points)

Deadline: May 10$^{\text{th}}$, before the lecture. The websites for lecture and exercises can be found at

`http://www.or.uni-bonn.de/lectures/ss16/ss16.html`

In case of any questions feel free to contact me at saccardi@or.uni-bonn.de.