Prof. Dr. Stephan Held
Paula Heinz, M. Sc.

Programming Exercise 2

Exercise P.1. Implement a program that computes the following items for a set of points in the plane $\mathbb{R}^2$ w.r.t. ℓ_1-distances:

(a) the bounding box netlength with a runtime of $O(n)$,

(b) the clique netlength with a runtime of $O(n \log n)$,

(c) the star netlength with a runtime of $O(n \log n)$,

(d) the length of a minimum spanning tree with a runtime of $O(n^2)$,

(e) a Steiner tree according to the following variant of the Steiner tree approximation algorithm from the lecture with a runtime of $O(n^3)$:

Choose $t \in T$ arbitrarily;
$Y := (\{t\}, \emptyset), S := T \setminus \{t\}$
while $S \neq \emptyset$ **do**
    Choose $s \in S$ with minimum $dist(s, Y)$
    if $E(Y) = \emptyset$ **then**
        $Y := (\{t, s\}, \{\{t, s\}\})$
    else
        Let $\{u, w\} \in E(Y)$ be an edge which minimizes $dist(s, SP(u, w))$
        $v := \arg\min\{dist(s, v) \mid v \in SP(u, w)\}$
        $Y := (V(Y) \cup \{v\} \cup \{s\}, E(Y) \setminus \{\{u, w\}\} \cup \{\{u, v\}, \{v, w\}, \{v, s\}\})$
    end if
    $S := S \setminus \{s\}$
end while

In this notation $SP(u, w) \subset \mathbb{R}^2$ is the area covered by shortest paths between u and w, and $dist(s, Y)$ is the minimum distance between s and the shortest path area $SP(u, w)$ of an edge $\{u, w\} \in E(Y)$.

The input should be read from an input pipe. Write the 5 numbers to the standard output. Fill non-computed numbers with '-1', if you are not able to implement all tasks. The data is given as a set of lines. Each line defines a pin through its x/y-coordinates, which are all integral. The following instance defines an example with four pins:

1	0
1	1
0	1
0	0

The program must be written in `C` or `C++` and must compile and run on Linux. You are allowed to use any any ISO `C` or `C++` standard including `C++20`. You can use any tool available in the standard library. Your program must compile with either Clang (any version $\geq$ 3.4.2) or Gcc (version $\geq$ 4.8.3) with `-Wall -Wextra -Wpedantic -Werror` and cannot link to any external library except the standard library. To achieve the maximum score, your program must not leak any memory and must be well documented.

Run your program on the following instances and create a table with the results:

`http://www.or.uni-bonn.de/~held/vlsi_design_ss08/SteinerInstances.tar.gz`

You can use the GeoSteiner program on these instances to compute optimum Steiner trees and compare with your results:

`http://www.geosteiner.com/`

(5+5+5+10+15 points)

Deadline: June 17, before the lecture. The websites for lecture and exercises can be found at:

`https://www.or.uni-bonn.de/lectures/ss25/chipss25_ex.html`

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