Research Institute for Discrete Mathematics
Chip Design
Summer term 2015

Prof. Dr. Jens Vygen
Markus Ahrens, M. Sc.

Programming Exercise 1

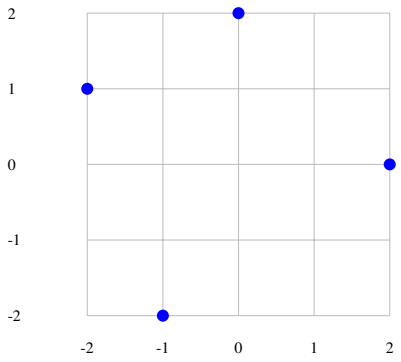
Implement the STEINER TREE APPROXIMATION ALGORITHM from the lecture for rectilinear instances in the plane. The running time must be $\mathcal{O}(|T|^2)$ and the program call must be

PROGRAM <INPUTFILE> <OUTPUTFILE>

The source code must be written in C or C++ and compile with GCC on Linux. It should be well documented [1]. You are allowed to use standard headers including the STL, but no other external libraries.

Input: The input file is a text file containing the number of terminals $|T|$ as its first line, followed by $|T|$ lines containing the x and y coordinates of the terminals. All occuring coordinates will be integers in the interval $[-2147483648, 2147483647]$. Coordinate pairs of terminals do not need to be distinct. An example input file looks like this:

```
5
-2 1
-1 -2
0 2
2 0
2 0
```



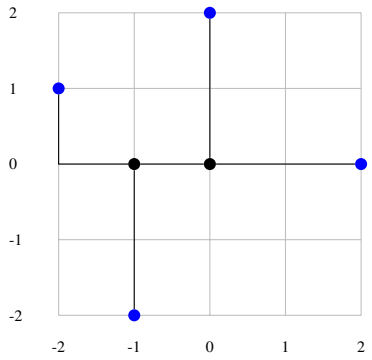
Output: The output file must start with a line containing the number of terminals and the total number of vertices (=: n) of the output tree. The second line must contain the length of the tree only. This should then be followed by n lines containing the indices and coordinates of the nodes, where the indices are distinct integers in $\{0, ..., n-1\}$ and all coordinates are integers in the interval $[-2147483648, 2147483647]$. The indices do not need to be given in sorted order, but the i-th terminal in the node listing in the input file must get index $i-1$ $(i = 1, ..., |T|)$.

The next $n-1$ lines should then describe the edges of the tree, given as a pair of indices of its endpoints. The output must describe a tree, and terminals with identical coordinates

[1] This can be achieved by using comments and – much more importantly – self-documenting code.

can be connected by edges of length 0. Diagonal edges are allowed, but their length will be counted as the ℓ_1-distance of their endpoints. An example output file looks like this:

```
5 7
9
0 -2 1
1 -1 -2
2 0 2
3 2 0
4 2 0
6 -1 0
5 0 0
1 6
6 0
5 6
2 5
5 3
3 4
```



Test instances will be provided on the website of the exercise class

http://www.or.uni-bonn.de/lectures/ss15/chipss15_ex.html.

The complete source code must be sent to *ahrens@or.uni-bonn.de* until

Thursday, May 14, 23:59h.

Since the algorithm is quite generic, give a short description of your implementation. The number of achieved points depends on code quality, running time and quality of the result (i.e. length of the tree).

(20 points)

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