Research Institute for Discrete Mathematics
Chip Design
Summer term 2013

Prof. Dr. S. Hougardy
D. Rotter, M. Sc.

Programming exercise 2

Implement an algorithm which computes the minimum perimeter of a rectangular chip image in which a feasible placement exists for a given set of n rectangles (without rotations).
Your program should enumerate all sequence pairs (Lemma 24) in order to compute $X_{i,j}$, $Y_{i,j}$ $(1 \leq i < j \leq n)$ as in Theorem 21 and evaluate them as in Theorem 25.

The source code must be written in C or C++ and has to compile with a GNU-compiler (gcc or g++) on linux. You are allowed to use standard headers including the STL. Your implementation should run in $\mathcal{O}((n!)^2 \cdot n \log n)$ time. The source-code should be well commented.

Input The first line contains a number $n \in \mathbb{N}$ specifying the number of rectangles to be placed. The remaining n lines contain two numbers specifying the widths and heights of the rectangles.
All positions of rectangles of the test instances will be integers and all coordinates can be represented as `long int`.

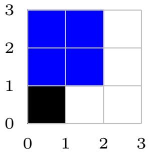
Example:
An instance with two squares with edge length 1 and 2
would be encoded as follows:

```
2
1 1
2 2
```

Output The output must consist of $n+1$ lines. The first line consists of two numbers specifying width and height of the computed chip area. The remaining n lines encode the positions of the lower left corners of the rectangles. The $i+1$st line consists of two numbers specifying x- and y- coordinate of the lower left corner of the rectangle corresponding to the $i+1$st line of the input file $(i = 1, \ldots, n)$.

Example:
The plotted solution for the example instance can be
encoded as follows:

```
2 3
0 0
0 1
```



Test instances will be provided on the website of the exercise classes.
The complete source code should be sent by e-mail to *rotter@or.uni-bonn.de* until ***Thursday, June 27th.***

(32 points)