1) Implement the $O(n^3)$-time algorithm from exercise 2.1 to compute the bounding box net length for a set of sets of rectangles.

The implementation must be done either in the C++ or C programming language respecting the C/C++ standard from 1999. You can easily achieve this by using the GNU-compiler (gcc or g++) and by including only standard headers (including the STL).

The input should be read either from an input pipe or directly from a file. The input format is as follows. The first line contains a number $n \in \mathbb{N}$ specifying the number of sets of rectangles. Then $n$ block are following, each containing the information for one set. The first line of each block contains the number of rectangles in the corresponding set. The next $k$ lines contain the four coordinates of one rectangle: $x_{\min}, y_{\min}, x_{\max}, y_{\max}$.

The values define the rectangle $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \subset \mathbb{R}^2$. Note that the rectangles might be 2-, 1- or 0-dimensional and might overlap other rectangles.

In the following example there are two sets of rectangles, the first one containing three, the second one two rectangles:

```
2
3
0 0 3 4
3 3 5 4
1 2 5 2
2
4 0 6 1
3 4 5 5
```

The program should write the result (coordinates of a bounding box rectangle and the bounding box net length) to the standard output. Here for example:

```
bounding box: 4 1 4 2
bounding box netlength: 1
```

More examples can be found on the web page of the exercises class. (12 points)
2) Let \( r \) be a pin and \( \alpha > 0 \).

- If \( r \) is a logical source, replace each signal \((a, s, \zeta, \xi) \in \Sigma(r)\) by \((a + \alpha, s, \zeta, \xi)\), i.e. all signals start \( \alpha \) later.
- If \( r \) is no logical source, replace delay\(_{(p,r),z}(\xi, s)\) by delay\(_{(p,r),z}(\xi, s) + \alpha\) for all \( p \in \delta^{-}(r), z, \zeta \) and \( s \), i.e. all signals at \( r \) are additionally delayed by \( \alpha \).

Let \( \text{slack}_{late}(r) \) be the slack at \( r \) according to the original information. For a logical sink \( q \) denote by \( \Sigma(q) \) and \( \Sigma'(q) \) the signals at \( q \) calculated using the original and the modified delays, respectively.

Prove that for each logical sink \( q \) and all \((a', s, \zeta, \xi) \in \Sigma'(q)\) there exists an \( a \in \mathbb{R} \) with \((a, s, \zeta, \xi) \in \Sigma(q)\) and

\[
a' \leq \max\{a, \text{rat}_{late}(q, s, \zeta, \xi)\}
\]

if and only if \( \alpha \leq \text{slack}_{late}(r) \). (4 points)

3) Let \( Y \) be a Steiner tree for a terminal set \( T \) in which all leaves are terminals. Prove:

- (a) \(|\{v \in V(Y) \setminus T : |\delta_{Y}(v)| > 2\}| \leq |T| - 2\).
- (b) \( \sum_{v \in T}(|\delta_{Y}(v)| - 1) = k - 1 \), where \( k \) is the number of full components in \( T \).

(4 points) 

4) Let \( T \) be an instance of the Rectilinear Steiner Tree Problem and \( r \in T \).

For a rectilinear Steiner tree \( Y \) for \( T \) we denote by \( l(Y) \) the maximum length of a path from \( r \) to an element of \( T \setminus \{r\} \) in \( Y \).

- Describe an instance in which no shortest Steiner tree minimizes \( l(Y) \) and no Steiner tree minimizing \( l(Y) \) is shortest.

- Consider the problem of finding a shortest Steiner tree for which \( l(Y) \) is minimum among all shortest Steiner trees. Is there always a tree with these properties which is a subgraph of the Hanan grid?

(4 points)

---

**Deadline for exercise 1**: May 16 (by e-mail to massberg@or.uni-bonn.de).

**Deadline for exercises 2-3**: May 4 before the lecture (12.15 pm).