

# Programming Exercises 1

## Exercise 1:

Implement the Edmond's cardinality matching algorithm.

(50 points)

## Guidelines:

1. The input file encodes a graph  $G = (V, E)$  and consists of  $m + 1$  lines. Line 1 consists of a single number  $n := |V(G)|$ , and each of the following lines consists of two numbers  $i j$  with  $i, j \in [0, \dots, n - 1]$  which represent an edge  $e = \{i, j\} \in E(G)$ .

Example: A  $K_3$  would be encoded that way:

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

2. The output file consists of  $\nu(G) + 1$  lines, where line 1 encodes two numbers  $n \nu(G)$  and the following edges encode the matching edges, again encoded by two numbers  $i j$  with  $i, j \in [0, \dots, n - 1]$  as above.

Example: The output of the program on a  $K_3$  would look like:

```
3 1
2 0
```

3. Please use the following naming conventions simply append `.out` to the input file to name the output file (If an input file is called `hugo` please write the output to `hugo.out`), i.e. the program will be called with the syntax `program hugo` and produce a file `hugo.out`.
4. Use ISO 99 C/C++.
5. The program has to compile with `g++/gcc` version 4.5 on linux.
6. 3rd party libraries (except STL and `stddef/stdlib/stdio` ) are not allowed.
7. Do not use nested directories. Please provide a Makefile (I will use `gnu make 3.81`)
8. See also a basic example on the net:  
[http://www.or.uni-bonn.de/lectures/ws10/co\\_uebung\\_ws10.html](http://www.or.uni-bonn.de/lectures/ws10/co_uebung_ws10.html)
9. In doubt: please ask.

**Deadline:** Monday, 10th January, 11:59 pm CET by e-mail to [struzyna@or.uni-bonn.de](mailto:struzyna@or.uni-bonn.de)