Summer term 2026
Prof. Dr. S. Held
Dr. U. Brenner

Research Institute for Discrete Mathematics
University of Bonn

Linear and Integer Optimization
Programming Exercise 2

Implement the (revised) simplex algorithm from the lecture. The algorithm will not get an initial solution but has to compute a starting basis by itself.
Your program should decide if an instance is infeasible, unbounded or can be solved optimally and should return a vector proving infeasibility or unboundedness in the first two cases.
If an instance can be solved optimally, your program should output optimum primal and dual solutions as well as the corresponding objective function value.
That way, correctness of the output can easily be verified. You may choose an index-strategy by yourself.
Implement two ways to compute the new vector x after an exchange step. In the first version, the new vector is computed by modifying the previous solution while in the second version the new vector is computed from the scratch by solving the corresponding equation system.
As input, the program expects a text-file obeying the specification described in Programming Exercise 1. Instances and the input reader can be reused.

The program has to be implemented in C++. Your submission must include all source files and a command that can be used to compile the program. The program will be invoked as `./fourier_motzkin input`. Your code must compile cleanly with the flags `-Wall -Wextra -Wpedantic -Werror -std=c++20` using `GCC 12+` or `Clang 15+`. No third-party libraries beyond the C++ standard library are permitted.
For reading in the instance, you can use a C++ program that can be found on the web page of the lecture.

For this programming exercise, you can get 30 points.

Due date: Tuesday, June 9, 16:00 s.t. The submission is done via the eCampus page of your exercise group.

Web page of the lecture: `https://www.or.uni-bonn.de/lectures/ss26/lgo_ss26.html`