Programming assignment 1

Use the PuLP library https://pypi.org/project/PuLP/ to solve the following problems. Documentation to PuLP can be found here: https://coin-or.github.io/pulp/main/index.html

Installation on linux and mac should be straightforward. On Windows and MacOS, if you use anaconda, you can use the following package: https://anaconda.org/conda-forge/pulp On windows, you can also proceed according to the following guide: https://learn.microsoft.com/en-us/windows/python/beginners and install necessary libraries using pip install. In case of problems, contact your TA!

A few hints

To surpress the messages from the solver, you can use GLPK(msg = 0) (if you use GLPK) or PULP_CBC_CMD(msg=False) (if you use CBC) or a similar option for other solvers as a solver parameter of solve() method. You can use function value() to evaluate any expression involving the variables in the optimal solution. If you need many variables, it might be good to introduce them using an array of their names using dicts() method and use the returned dictionary to access them.

Instructions for preparing your submission

Submitted file should be called main.py and should implement functions ex1(), ex2(), and ex3() solving the respective exercises. Download the attached template file and fill it with your code. At shubmission time, you will receive a report checking whether your code returns the output in the required format. This test does not check the correctness of the solution, it only checks whether the format of the solution is acceptable by the grading script. Make sure that all checks are passed, you can resubmit multiple times. If your code does not respect the required format, grading script will not grade it and you will receive 0 points.

Part 1 (10 points)

Solve the following program:

```
       min 122x + 143y
subject to:
         x ≥ -10
         y ≤  10
  3x +  2y ≤  10
 12x + 14y ≥ -12.5
  2x +  3y ≥   3
  5x -  6y ≥-100
```

The constraints are ordered: the first one (x ≥ -10) has index 1, the last one (5x - 6y ≥-100) has index 6.

Required output example:

```
    retval["x"] = 0.1
    retval["y"] = -2.3
    retval["tight_constraints"] = [ 1, 2, 4]
    # Don't forget to return your dictionary!
    return retval
```

So, your program should find the optimal solution, determine its objective value, and report the index of each tight constraint. The example above claims that the tight constraints are x ≥ -10, y ≤ 10, and 12x + 14y ≥ -12.5.

Part 2 (10 points)

Find an optimal mixed strategy of the following game: Both players choose independently a single integer from 1 to 6. Then, the numbers are compared:

- If they are equal, there is a draw
- If they differ by 1, the player who played the smaller number gets 2EUR from the other player
- If they differ by ≥2, the player who played the larger number gets 1EUR from the other player

Note that the game is symmetric and the same strategy is optimal for both players.

Part 3 (10 points)

On some imaginary island, there are 69 companies and there are bilateral contracts between them. The monarch of the island would like to inspect validity of each of these contracts during a single large event. The monarch requires two representatives to represent each contract relationship (they can be both from the same party of the contract or each from a different one). This is of course satisfied by each company sending a single representative, which would require involvement of 69 representatives in total. However, the companies want to find a solution which minimizes the total number of representatives who need to attend the event.

Input file hw1-03.txt contains information about the contracts. Each line corresponds to a single contract and contains identifiers (1-69) of both involved parties separated by a space.

Important: It is not possible to send an arbitrary fraction of a representative. However, it is enough to solve the LP relaxation since it already gives an integral solution. Reasons for this will be explained later in the class.