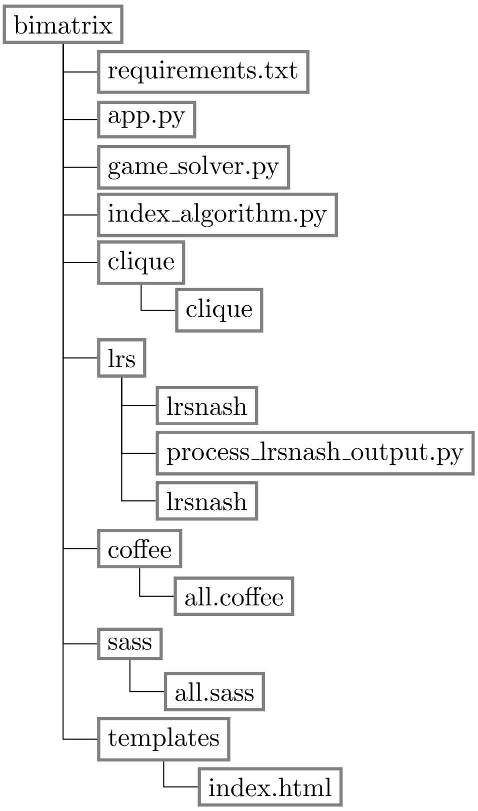
Bimatrix Game Solver Program

Installation and usage instructions

Folder structure

We present the folder structure followed by explanations of the keys files of the program. Folder structure is depicted in the following figure.



Key files descriptions

- **requirements.txt** list of software dependencies to install in order to run the web server.
- **app.py** contains the code for the web server (Flask Python application).
- **game_solver.py** Python script that executes the game solving algorithm.
- **index_algorithm.py** contains the core of the index computation algorithm.
- **lrs/lrsnash** executable file for the lrsnash program implemented by David Avis.

- **lrs/lrs_input** main input file of the program (contains matrix dimensions and two payoff matrices)
- **lrs/process_lrsnash_output.py** Python code written by Raul Savani that parses the output of 'lrsnash' and creates the input to 'clique'.
- **clique/clique** executable file for a clique enumeration program implemented by Bernhard von Stengel.
- **coffee/all.coffee** JavaScript code for the web page (written in CoffeeScript).
- **sass/all.sass** CSS code (styling) for web page (written in SASS)
- **templates/index.html** HTML code for the web page.

Installation and usage

First we need to compile *lrsnash* and *clique*. This can be done using the following commands.

```
gcc -O3 -o lrs/lrsnash lrs/lrsnash.c lrs/lrsnashlib.c lrs/lrslib.c lrs/lrsmp.c
gcc -O3 -o clique/clique clique/coclique3.c
```

There are two options to use the program.

1. Through the command line.
2. Through a web browser using web interface.

Using the command line

In order to run through the command line one should create a file named *lrsnash_input*, place it in the *lrs* directory and execute the following command:

```
python game_solver.py
```

The structure of *lrsnash_input* should be as follows.

1. Matrix dimensions separated by space.
2. Blank line.
3. Payoff matrix to player 1 separated by spaces.
4. Blank line.
5. Payoff matrix to player 2 separated by spaces.

Example for an *lrsnash_input* file for 2 × 3 bimatrix game is the following.

```
2 3

1 2 3
4 5 6

7 8 9
10 11 12
```

Output will be printed to the console. Example of output for the above game will be the following.

```
INPUT:
Payoff matrix to player 1:
[[ 1.  2.  3.]
 [ 4.  5.  6.]]
Payoff matrix to player 2:
[[  7.   8.   9.]
 [ 10.  11.  12.]]

OUTPUT:
EXTREME EQUILIBRIA
   Equilibrium number: 1
     Player 1
       Strategy number: x1
       Distribution: [0, 1]
       Payoff: 6
     Player 2
       Strategy number: y1
       Distribution: [0, 0, 1]
       Payoff: 12

EQUILIBRIUM COMPONENTS
   Component number: 1
     Nash subsets:
       [x1] X [y1]
     Extreme Equilibria
       Number: 1 , Lex-index: 1.0
```

Running the web server

A live running version of the program is available at:
`https://bimatrix.herokuapp.com`
In order to run the server locally we need to install all its components using the following command (requires up to date version of Python to be already installed).

```
pip install -r requirements.txt
```

Next we can run the local server using the following command:

```
python app.py
```

Application can be accessed with the following local url.
`http://127.0.0.1:5000`