Algorithmic Game Theory

Winter Term 2019 / 2020

Prof. Dr. Martin Hoefer, Dr. Daniel Schmand

JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Algorithmen und Komplexität
Institut für Informatik

Exercise Sheet 11

Publication: Jan 28, 2020
Solutions Due: Feb 04, 2020

Please hand in your solutions until Tuesday, February 04, 10:15h, in H9 or in the letterbox between rooms 114 and 115, R.M.S. 11-15.

This exercise sheet contains a programming exercise. Until Tuesday, February 04, 10:15, please send the output files for the instances `input10.txt` and `input100.txt` by mail to Daniel and hand in the solution to exercise 11.1b). Additionally, explain your working algorithm to Daniel until Thursday, February 06, 18:00, at the latest.

Exercise 11.1. (8+2 Points)

a) Implement the Top-Trading-Cycle algorithm from the lecture. For this exercise we will only provide input files and leave it to you to solve the problem in the way you like it. For this programming exercise, you need to think about how you store the preference orders, and you need to design your own subroutine to calculate all cycles in a current round. You can use your favorite programming language. For both the input and output, we will refer to the agents by numbers $0, \dots, n-1$. You can assume $n \geq 1$. For this exercise, we expect you to solve instances with $n \leq 1000$ in reasonable time, i.e. within in some seconds. For completeness, we also provide larger instances on the website. Please write your solution into an output file as specified below.

The input file is structured as follows: The first line of the input specifies the number of agents, n. After the first, n lines follow. The i-th line of the file contains a list of integers (from 0 to $n-1$) that specifies the preference list of agent $i-2$: The first entry of line i is the most preferred house of agent $i-2$, the second entry gives the second most preferred house and so on. Equivalently, the houses are sorted decreasingly according to the preference order of agent $i-2$.

Your output file should contain n lines, where the i-th line specifies the house that is allocated to agent $i-1$.

You can find example files for both input and output on the website.

Hint: In Java, you can read input files with the Scanner-class (located in `java.io`), in Python you can do this with the `open`-method.

b) What is the asymptotic worst case running time of your algorithm? Reminder: The worst-case running time of an algorithm specifies the number of operations that an algorithm requires in dependence of the input size b.

Exercise 11.2. (2+2 Points)

a) Prove that the ~~Random~~ Serial Dictatorship (RSD) algorithm is incentive compatible for every a-priori fixed permutation π of players.

b) Show that there is an instance and a permutation such that the outcome of the RSD algorithm is not in the core.

Exercise 11.3. (4 Points)

Prove that the matching mechanism with priority lists for kidney exchange is incentive compatible. Here, we assume that players are the patient-donor pairs. It is sufficient to show that an unmatched player cannot get included into the matching by not reporting a compability.

The exercise sheets and more information about the course can be found at `http://algo.cs.uni-frankfurt.de/lehre/agt/winter1920/agt1920.shtml`

Email: `mhoefer@cs.uni-frankfurt.de, schmand@em.uni-frankfurt.de`