SUFE: Algorithmic Game Theory Summer School	Problem Set 1a
Due: July 19, 2017, 23:59pm	*Submit solutions to any 3 problems out of 4.*

Please send your solutions to **nikolai@mail.shufe.edu.cn**. You may submit typed in latex pdfs, or scans/photos of your handwritten solutions.

1. **Path Zero-Sum Games**: Players 1 and 2 are playing a zero-sum game with payoff matrices $(A^{1,2}, A^{2,1})$; at the same time players 1 and 3 are playing another zero-sum game with payoff matrices $(A^{1,3}, A^{3,1})$. Instead of using different strategies in the two games, player 1 must use the same strategy in both games. Show that there exists a Nash equilibrium in this game using linear programming duality.

2. **Nash Equilibrium Computation:** Suppose (R, C) is a $n\times m$ two-player game, where the entries of R, C are rational numbers in $[0, 1]$. Denote by $[n] = \{1, \ldots, n\}$ and $[m] = \{1, \ldots, m\}$ the pure strategies of the row and column players respectively.

   (a) Construct a polynomial-time algorithm that, given a game (R, C) as above and two sets $S \subseteq [n]$ and $T \subseteq [m]$, computes a Nash equilibrium (x, y) where the mixed strategy of the row player is supported on S and the mixed strategy y of the column player is supported on T, if a Nash equilibrium with supports S and T exists.

   (b) Construct a $O(2^{n+m} \cdot \text{poly}(n, m, b)$ algorithm that, given a game (R, C) as above, whose payoff entries are rational numbers of bit complexity b, computes a Nash equilibrium of the game.

   (c) **(Bonus)** Suppose that you are given a game (R, C) as above, whose payoff entries are rational numbers of bit complexity b, along with an $\hat{\epsilon}$-approximate Nash equilibrium $(\hat{x}, \hat{y})$ of this game for $\hat{\epsilon} = 1/2^{(m\cdot n\cdot b)^{2017}}$. In particular, $(\hat{x}, \hat{y})$ satisfies:

   $$\forall i, \text{s.t.} \hat{x}_i > 0 : e_i^{\mathrm{T}} R\hat{y} \geq e_k^{\mathrm{T}} R\hat{y} - \epsilon, \forall k; \tag{1}$$
   $$\forall j, \text{s.t.} \hat{y}_j > 0 : \hat{x}^{\mathrm{T}} Ce_j \geq \hat{x}^{\mathrm{T}} Ce_k - \epsilon, \forall k. \tag{2}$$

   Given (R, C) and $(\hat{x}, \hat{y})$ as above provide a polynomial-time algorithm that computes an exact Nash equilibrium (x, y).

3. **Player Exchangeability:** In class we saw that every game with a finite number of players and a finite number of strategies per player has a Nash equilibrium. Consider now an n-player game in which the first two players are "exchangeable" in the following sense:

   - they have the same strategy sets $S_1 = S_2$;
   - their utility functions u_1, u_2 satisfy the following property: for all pure strategy profiles $(s_1, s_2, s_3, \ldots, s_n)$, $u_1(s_1, s_2, s_3, \ldots, s_n) = u_2(s_2, s_1, s_3, \ldots, s_n)$.

   Show that any game satisfying the above exchangeability property has a Nash equilibrium $(x_1, x_2, x_3, \ldots, x_n)$ such that $x_1 = x_2$.

4. **Splitting the Rent:** Suppose that 3 roommates want to split among themselves the rooms and the \$3,000 rent of a 3-bedroom apartment. Let's call the rooms 'R,' 'G,' and 'B' for red, green and blue, and the roommates 'C,' 'N,' and 'Z' for Costis, Nick, and Zhihao. Note that the rooms are not identical and different roommates may value different features of the rooms differently. Your goal is to come up with an assignment of C, N and Z to rooms R, G, and B together with a split of the \$3,000 of rent into how much of it C, N and Z should pay so that everyone is almost happy. An assignment of rooms $r_{\mathrm{C}}, r_{\mathrm{N}}, r_{\mathrm{Z}}$ and payments $p_{\mathrm{C}}, p_{\mathrm{N}}, p_{\mathrm{Z}}$, where $p_{\mathrm{C}}+p_{\mathrm{N}}+p_{\mathrm{Z}} = 3000$, make everyone almost happy, if there are no $i, j \in \{\mathrm{C}, \mathrm{N}, \mathrm{Z}\}$, $i \neq j$, such that i prefers room r_j at price $p_j + 1$ to room r_i at price p_i. Come up with an algorithm that computes a room assignment and rent

division that makes every roommate almost happy. Your algorithm can interact with C, N and Z, making them queries of the following form: "given prices p_{R}, p_{G}, p_{B} such that $p_{\mathrm{R}}+p_{\mathrm{G}}+p_{\mathrm{B}} = 3000$, which room do you prefer?" We will assume that whenever some room has 0 price, it is preferred by any one of C, N, Z to a room with a non-zero price, so your algorithm should work under this assumption.