**CS 332 Fall 2025**

Project #3

**Koshi Harashima, Ben Cole**

Due Date: 11/5, 2025

## Intro

In this project you will consider using learning algorithms to play in repeated games, i.e., the same game repeated many times. You can consider repeated bimatrix games, repeated first-price auctions, or repeated generalized second price auctions (defined below). You are free to choose whether to consider full feedback or partial feedback versions of the learning problem. It is recommended to consider games of full information, e.g., for auctions, assume that each bidder has a fixed value that is the same in every round. You are looking for generalizable conclusions, so you should vary the settings you consider to make sure your observations are robust. For example, in bimatrix games, vary the exact payoffs. In auctions, vary the values of the bidders. Demonstrate your results using Monte Carlo simulations.
Complete the following parts and prepare a project report according to the Project Report Guidelines. Submit your project reports on canvas. Do not include your names on your submission. For group projects, you must join a Project Group before submitting.

## Part 1: Outcomes from No-regret Learning in Games

In a family of games of your choice, consider playing learning algorithms against each other, i.e., each player's action should be determined by a learning algorithm. Here are some questions you could answer. What outcomes do you observe? Do they converge to a Nash equilibrium? In games with multiple Nash equilibrium, which one is converged to? Are some learning algorithms better than others? Is there an interplay between the learning rate and how well the learning algorithms do? (For example, you could consider a Nash equilibrium in the meta-game where the the players are choosing learning rates, and then playing learning algorithms against each other with these rates.)
Recall: a "no-regret" learning algorithm is one where when the learning rate is tuned appropriately, the average per-round regret vanishes with the number of rounds, e.g., exponential weights.

## Part 2: Manipulability of No-regret Learners in Games

Consider playing in the games of your study in Part I, and fix a no-regret learning algorithm of your opponent. Devise an algorithm that is a good response to this learning algorithm. For example, perhaps you can force the no-regret learning algorithm into accepting an outcome that is high-payoff for you even if it is not a Nash equilibrium in the stage game. As an example, consider the first-price auction with bidders with value 90 and 30. As we saw, there is no Nash equilibrium where the 30-valued player wins. But, if the 90-valued player is playing a specific no-regret learning algorithm, can you devise a strategy where the 30-valued player can sometimes win at a low price?

### Generalized Second Price Auction

The generalized second-price auction is used by Google to sell advertisements on its search results page. (Note: truthful bidding is not a dominant strategy in the generalized second-price auction.) The following model is useful in understanding this auction. The advertisement positions are ordered, and the ads shown in higher positions are more likely to be clicked on than ads shown in lower positions. Suppose there are n bidders and n positions. Denote the click probabilities of the positions by $w_1 \geq ... \geq w_n$. If the advertisers' bids (after sorting) are $b_1 \geq ... \geq b_n$ then, advertisers are assigned in this order to slots. If an advertiser is clicked on, then the advertiser pays the bid of the ad in the slow below (or zero for the lowest bidding advertiser). Thus, an advertiser's utility with value v, who is bids bi and is in the ith highest slot, is $(v - bi + 1)wi$.