Trading Bot Competition: Problem Statement

Overview:

In this competition, each participant will program a trading bot that competes against other bots in a multi-round bidding game. The objective is to develop the most effective bidding strategy to maximize profit over several rounds.

Please go through the below doc in detail. Feel free to post any questions you have on the guild whatsapp group - https://chat.whatsapp.com/LK9Ga86Cxve0Bmc4aM6o5t
You can also drop an email at cs21b024@smail.iitm.ac.in, or cs21b082@smail.iitm.ac.in, with the subject “Quant PS1”.
We will try our best to respond within a day.

Game Setup:

- **Number of Players**: n (each player is a bot developed by participants)
- **Number of Rounds**: t (~10^3)
- **Initial Condition**: At the start of each round, player i is assigned a value x_i, which is randomly drawn from a predefined distribution.
- **Maximum Bid Value:** 100
- **Starting Capital:** Will be varied

Gameplay:

1. In each round, every player is given a value x_i (All x_i's are independent of each other i.e. they need not be the same for all players).
2. Each x_i is drawn from a distribution (specified in the next part).
3. Every player will submit a bid. Bids can be fractional as well, but must lie in the range [0,maximum value].
4. The player with the highest bid wins the auction. If multiple players are tied for the highest bid, all of them win the auction.
5. Every player receives a payoff, which is explained below.
6. All other players receive a zero payoff for that round.
7. Players start off with a fixed amount of capital, which gets updated according to their payoffs i.e. **new capital = old capital + payoff**.
8. At the start of each round, players will be given the highest and second highest bids of the previous 100 rounds, as well as the amount of capital they have left, and the number of players participating in that round of the auction.
9. Once a bot runs out of capital, it will no longer be able to participate in the future rounds in the auction.

10. If a player makes an illegal bid (bid > capital available, or bid does not lie in [0,maximum_value]), then that player's bid for the round will automatically be set to 0.

Auction Variations:

1. **Variation 1:**
    - The value x_i for each player, not necessarily the same for all, is drawn from a **uniform random distribution** over the range [0,maximum value]. All players know that the values are uniformly distributed.
    - The **payoff** for the **winners(player i)** is given by **Payoff = x_i - bid**, where **bid** is the **winning bid.**
2. **Variation 2:**
    - The value x_i for each player, not necessarily the same for all, is drawn from a **uniform random distribution** over the range [0,maximum value]. All players know that the values are uniformly distributed.
    - The **payoff** for the **winners** is given by **Payoff = X - bid**, where X is the **max value** amongst **all x_i**, **bid** is the **winning bid**.
3. **Variation 3:**
    - The value x_i for each player, not necessarily the same for all, is drawn from a **uniform random distribution** over the range [0,maximum value]. All players know that the values are uniformly distributed.
    - The **payoff** for the **winners** is given by **Payoff = X - bid**, where X is the **max value** amongst **all x_i**, **bid** is the **winning bid**.
    - In addition to this, the second-highest bidder will have to pay (50%) of what the winner earned i.e. for second-highest bidders **Payoff_second = -0.5*(X-bid)**
    - If **X-bid < 0,** Payoff_second = 0

Objective:

The goal is to maximize the total payoff over t rounds. Participants must design a strategy for their bot to decide on the optimal bid based on the information available during each round.

Additional Details:

- A **starter** code has been provided in the app. Make sure you only modify the file **Template.py.** The remaining files are **only meant for reference**, if you wish to understand how the auction is simulated. Details on how to use the code are provided in the file README.md
- We have also included 3 sample bots for you to get an idea of what we are expecting.
- Each player will only know their own x_i in each round, not the "value" of other players.
- The maximum possible bid value will also be specified ahead of the competition.

- Make sure that your bot does not take too long per round to run. If that happens, it may be removed from the auction. 1 second is a good upper bound to have in mind.
- Additionally, if it is noticed that the bot is hogging up a lot of memory, it will be discarded from the auction. Try to limit memory usage to below 100 mB.
- You may use any python library available. Explicitly mention the python libraries used in your report.
- Around 2-3 days before the deadline, we will hold a mock auction, where you can choose to submit your codes, and we will run a mini-auction on them. Relevant statistics such as net profit, variation of capital over time will be shared with you. The exact date of the mock auction will be announced on the guild group.

Deliverables:

Participants must submit their bot codes(**python files**), which follows the provided template, along with a brief report explaining their strategy.

A separate code is required for each variation. You can choose to submit codes for only some of the variations as well.

If your Rollno is OB24C420, then the submission for variation 1 should be named **OB24C420_1.py**.

SImilarly, the other submissions should be named OB24C420_2.py, OB24C420_3.py.

Add all python files to a folder called “Submissions” in your google drive, and submit the link for this folder in the following form:

https://forms.gle/9MfF49gfHp2naB3u7

Make sure to give view access to this folder.

The final submission deadline is **11th September, 2024 11:59 pm.**

Evaluation:

- All codes will be first tested against some sample bots coded by us, to check that it does not crash, or run out of capital very fast and performs as expected.
- All the bots that pass this preliminary check, will be divided randomly into groups of 20, and the auction will be simulated in these 20 groups.
- The above step will be repeated 3 times, to ensure that the effects of luck are nullified.

Metrics:

- We will mainly look at your report and the logical basis for your strategy.

- Net Profit by itself will not be a metric. It will be considered in addition to how your strategy adapts to different conditions.
- Robustness against different starting capitals.

Sample Run

Given below is a sample execution of 1 round of the auction between 3 bots.

- Bot 1 - Always bids 15 more than their own value.
- Bot 2 - Always bids 5 more than their own value.
- Bot 3 - Always bids half of their own value.

At the start of a round, the bots have a capital of **100,100,100** respectively.

In the round, the values obtained by the 3 bots are **30,50,60** respectively.
The bots bid **45,55 and 30** respectively.

Bot 2 wins the auction. Bot 1 has the second-highest bid.

Variation 1:

- Bot 2 gets a payoff of 50-55 = -5.
- Capital after the round:
    - Bot 1 - 100
    - Bot 2 - 95
    - Bot 3 - 100

Variation 2:

- Bot 2 gets a payoff of 60-55=+5
- Capital after the round:
    - Bot 1 - 100
    - Bot 2 - 105
    - Bot 3 - 100

Variation 3:

- Bot 2 gets a payoff of 60-55=+5
- Bot 1 gets a payoff of -0.5*5 = -2.5
    - Capital after the round:
    - Bot 1 - 97.5
    - Bot 2 - 105
    - Bot 3 - 100

Resources:

Following is a list of non-exhaustive resources for the above Problem set.
Introduction to Random Variables (Probability) -
https://www.investopedia.com/terms/r/random-variable.asp

Introduction to Python Programming : https://www.w3schools.com/python/python_intro.asp

Useful python module(Numpy) - https://www.w3schools.com/python/numpy/numpy_intro.asp

General Guidelines:

- Feel free to make extensive use of ChatGPT. We are trying to test the logic behind your strategy, not your coding skills.
- That being said, please indent the code properly, and add comments wherever possible, to make it easier for us to read it.

ALL THE BEST!