Assignment 3:
Bidding Strategies in Multiple Item Auctions

Implement a bidding agent who performs well in a complex auction environment of 3 continuous auctions running simultaneously.

Auction A

- Continuously sells small bundles of items by a second price sealed bid mechanism.
- Bundles have different sizes from 2 to 4 (uniformly distributed).
- Auction closes after 100 rounds. (I.e., approximately 300 items will be offered.)
- If no bids are placed for a bundle, there will be no transaction in the corresponding round and the bundle will be disposed.

Auction B

- Continuously buys specified bundles (also called *collections*) at fixed prices.
- Every round starts with the communication of the current prices for all collections.
- All incoming valid bids are processed simultaneously.
- Each transaction decreases a bundle's price by a constant Δ=5€ for the next round.
- The price of a bundle with no sales order will be increased by 5€ in the next round.
- In the case of many simultaneous orders on the same bundle, all bidder agents will receive the same price per collection. Example: 3 sales orders at an ask price of 100. Each order will be closed at a price of 95 (100+95+90)/3. In the next round, the price of the bundle will drop to 85.
- Auction B closes 40 rounds after termination of auction A.

Auction C

- Continuous exchange of single goods via a first price sealed bid mechanism.
- In each round, every agent is allowed to place 1 selling order for 1 item (of her inventory) with a reservation price.
- Orders are valid only for the current round, they will be automatically deleted afterwards.
- The auctioneer informs all agents about the current orders (item, price, ID of the selling agent).
- Every agent may then place bid(s) for any order(s).
- For each order, the highest bid will win - if it is valid (i.e., not below the reservation price). Tiebreaks are resolved randomly.
- Auction C closes 20 rounds after termination of auction A.

Items and Collections

Items {A, ..., K} are being sold in different quantities: Be p the probability of the occurrence of the scarcest item. Then, the probabilities are being distributed as follows:

$$\{A, B\}: 4\times p\,;\ \{C, D\}: 2\times p\,;\ \{E, F, G\}: 1\times p\,.$$

So, items of type A are on average four times as frequent as items of type E.

{J, K}: These items will be distributed to all bidder agents randomly at the start of a simulation. Each agent either receives a certain among x of J items or the same amount x of K items. There is an even distribution of J and K items. Those items are traded only in auction C.

Your main task will be to buy bundles on auction A, build profitable collections out of your inventory and sell them - at the right time - on auction B. In total 16 different collections may be built. The typical start prices (in parentheses) for these collections are:

AA (200), AAA (300), AAAA (400), AAB (200), AJK (200),
BB (50),
CCCDDD (1200), CCDDAA (800), CCDDBB (600),
EEEEEF (1600), EEEEF (800), EEEF (400), EEF (200),
FF (100), FJK (300),
ABCDEFJK (1400).

The smallest collection comprises just 2 items, the largest collection is a set of 8 items. Item G is not used in any collection.

Competitions

Each competition consists of a sequence of contests. The first (and each other odd-numbered) contest is always the *FIXED Contest*, followed by the *Random* Contest (which is also conducted in all other even-numbered rounds.)

In each contest, your bidding agent is initialized with a budget of 10000€ and an initial inventory of {J,K} items.

1. *FIXED Contest* starts with a commonly known, fixed sequence of bundles of items at auction A (according to Benchmark_bundles.txt on ISIS). Start prices at auction B are exactly as being specified above.
2. *RANDOM Contest* starts with a random sequence of bundles of items at auction A. Start prices at auction B may deviate from the specification above by a margin of 50%.

At the end of each *Contest*, every item in an agent's repository will be valued -20€ on the current budget.

Process and results are being recorded in log files (a separate log file for each contest).

Messages and Failures

1. Each auction round ends by informing all participating agents about the auction's outcome. The inform message from the auctioneer contains the sold item(s) and price(s), but no information about the winning bidders' IDs.
2. Buy orders exceeding the current cash will be executed (the auctioneers do not check your balanced budget). Therefore, your bidding strategy may result in a temporary negative balance.
3. Any negative balance of items at the end of a competition will be punished hard, by a penalty of 10.000€ per missing item.
4. Inconsistent sell orders (of items not available in the inventory) will be ignored.
5. If more than 1 sell order is submitted by an agent to auction C, one of them will arbitrarily be chosen.

Your task

1. stage (20%)
- Implement a bidding agent with a reasonable strategy. Your *simple bidding agent* will serve as a benchmark in later simulations, also for the other participating groups. That is, your code will be made available via ISIS.
- In order to support the sharing of code, you are obliged to modify only the BidderBean.java for your *simple bidding agent*.
- Evaluate your agent (locally, by running the auction server). You may implement or clone some more bidding agents for that task).
- First contest: auction with one bidder agent of every group.
- Short description of your bidding strategy (1 page of text).

2. stage (80%)
- Implement more sophisticated bidding strategies.
- Set up some experiments for finding the optimal bidding strategy. You may run some simulations on your own and use the bidder agents of the other groups of stage 1.
- Second contest: Participate with your bidding agent in some competitions that are being conducted on a dedicated server. Those contests will be started at specific times in regular intervals.
- Write a short paper describing your approach: How did your optimal bidding strategy evolve?
- Provide a screencast of your approach and results.

Timeline:

2021-02-05 (Friday) First stage competitions. Your simple bidding agent participates (as a competitor of the other groups' agents) in the first competition.
Upload your simple bidding agent code into *Group forum for assignment 3*.

2020-02-19 (Friday) Second stage competitions. In the final competition, your intelligent bidding agent will compete against the other groups. Some simulations may contain additional benchmark agents from the AOT chair (possibly supplemented by some bidding agents from the first competition).

2021-02-26 Upload into the ISIS activity *assignment 3*:
- a link to your screencast
- your code (or a link to your code repository)
- your short paper

Code repository:

JIAC code repositories: https://www.jiac.de/agent-frameworks/jiac-v/ (use the additional code base which can be accessed from the ISIS course page)

Server running on (to be announced later) Check the observer at (to be announced later).

Each round takes 1 second, resulting in a duration of approximately 5 minutes for 1 contest.

Auctions start each 15 minutes (alternating FIXED and RANDOM).