**Problem Set 7A (due 11:59 pm, 21 December 2023)**

**Wealth Game of New York Stock Market**

Write a Python program modeling agents using Wealth Game strategies and payoffs to invest in the time series of New York Stock Market. Fix the number of strategies held by each agent to , and the memory size . Below are the suggested steps:

Formulation of the market

1. Download the time series of New York Stock Market (^NYA) from 1 January 2009 to 31 December 2018. This will form the time series of the prices .
2. Set the market impact factor to Starting from , the time series of the transaction price will be generated using
3. Initialize the input state at to 0 (that is, 00 for ).
4. The input states to the strategies used by the agents are the -bit histories of the price change (each bit being equal to 1 for price rise and 0 for a price drop), and the outputs are the buy/hold/sell decisions of the strategies (represented as respectively. Starting from calculate the input states of the strategies according to

where for price rise or drop from to respectively. (Hint: Remember to specify “dtype = int” for the elements of the state array.)

Construction of the strategies

1. Prepare the strategies of the agents as a three-dimensional array with size that is, darray[i, j, mu] is the output of strategy j of agent i when the market state is mu. The outputs are for buy/hold/sell decisions respectively with equal probability. (Hint: use random.randint(-1, 2, size=<size>)).
2. Initialize the virtual wealth of the strategies to

Zeroth step of the Wealth Game

1. Initialize the real wealth of the agents to
2. Since the virtual wealth of all strategies are the same at select one of the strategies of each agent randomly (Hint: use random.randint()).
3. Set the initial state of the game to be
4. Denoting as the decision of agent i, set the real positions of the agents to (Remark: Since all agents start with position 0 before the game and are given 5 shares of the stock, the restriction due to the maximum position is not relevant.)
5. Similarly, denoting as the virtual decision of strategy j of agent i, set the virtual positions of the strategies to

Entering the Wealth Game

Starting from until the end of the game, perform the following:

1. Compute the price change
2. For each agent, select the strategy with the highest virtual wealth and output the decision following that strategy. (To speed up the program, you may simply use the command argmax() neglecting the chance of having two strategies with equal virtual wealth.) This allows you to update the real wealth of the agent by

and the real position by

Note that the max and min functions are used to restrict the positions to within the limit.

1. Similarly, update the virtual wealth and virtual positions of the strategies according to

Presentation of results

1. Plot a graph of the real wealth of the best 3, median 3, and worst 3 of the agents at the end of the game. For comparison, also plot the stock index multiplied by 5. (Hint: use argsort() to identify these agents.)
2. Report the real wealth of the 9 agents mentioned above. (a) Compute the gain relative to the initial wealth of the agents. (b) Compare the final wealth with the final value of the same number (=5) of stocks kept throughout this period.
3. Comment on the pros (if any) and cons (if any) of Wealth Game based on the above results. This is an open-ended question and I encourage you to report any observations you find interesting. For example, do you prefer to make frequent transactions aiming at high return by using strategies such as Wealth Game, or do you prefer to buy the stocks and keep them throughout the period? Are there particular periods that Wealth Game perform well? Or perform badly? Etc.