

## Task Bajt Trade

The goal of this task is to create a market simulation in which Agents participate, whose goal is to acquire as many diamonds as possible. We distinguish two basic types of Agents: Workers and Speculators. In case of any doubts, please refer to the assumptions section.

### Simulation Process

The simulation will take place in turns (also called days). We start the simulation from turn number 1. Each turn consists of the following stages performed in the following order. Each of the stages is described in detail in the appropriate section:

- 1) Workers decide whether to devote the day to learning or working.
  - a) The Worker learns. After learning, the day ends (no points below apply to him).
  - b) The Worker works.
    - i) The Worker, based on his strategies, produces items.
    - ii) The Worker places on the exchange sale offers of products he has produced. Workers' sales offers do not have a price, only the number and type of items they want to sell.
    - iii) The Worker places purchase offers. Similarly, purchase offers do not have a price, only the type and number of items.
- 2) Speculators enter the market with their purchase and sale offers. Their offers already have (in addition to the number and type of products) a price.
- 3) The Exchange performs the process of matching purchase offers with sales offers. Only Workers' offers will be matched with Speculators' offers, thus determining the transaction price. Therefore, a Worker will not sell/buy anything to another Worker. Similarly, the Speculator.
- 4) After the realization of matched transactions, the Exchange buys unsold products from Workers. Unfulfilled offers of Speculators (purchase and sale) and Workers (purchase) are cleared from the Exchange's memory. Therefore, each turn the Exchange starts without offers.
- 5) At the end of the day, Workers consume their items: food, clothes and tools, and used computer programs.

It is worth noting that a learning Agent does not consume any items at the end of the turn, including food.

### Workers

Each Worker starts the simulation with a fixed number of products (read from the input). Additionally, each Worker has permanently assigned a base productivity vector, which for each product determines how many units of this product the Worker is able to produce in one turn. The numbers in this vector will always be multiples of 100. During the simulation, this value will be modified by various bonuses (expressed in total percentage, e.g., +10%, +20%, -1%, -150%), so Workers will always produce a whole number of products per turn. If after adding bonuses (they can be negative) it turns out that we produce a negative number of items, we do not produce anything. For example: the base tool production is 200 and we have the following bonuses -10%, 20%, 40%. In a given turn, we will produce  $200 + 200 * (20 + 40 - 10)\% = 200 + 100 = 300$ .

In each turn, the Worker can either learn or work. Learning involves advancing in the current career path or changing the career path. Work involves production, sale, and purchase of products. If the Worker works in a given turn, he also consumes (loses) items in the following way:

- Consumes 100 units of food. If he has less, he consumes as much as he has, but it counts as if he did not eat anything that day.
- Consumes all his tools.
- Adds one day of wearing to any 100 clothes. The remaining clothes (if he has any) remain unchanged. A given piece of clothing will be completely worn out if it was worn as many days as its durability. If the Worker has fewer than 100 clothes, he adds a day of wearing to all he has.
- Consumes those computer programs that he used for production that day. The rest remain unchanged.

When does a worker learn and when does he work?

Each Worker has one of the following strategies, which determine what he does on a given day:

- Workaholic never learns, always works.
- Thrifty learns only when he has more than `limit_diamonds` diamonds, where `limit_diamonds` is the strategy parameter.
- Student always learns when he can afford to buy  $100 * \text{stock}$  units of food at a price equal to the arithmetic average of the average prices over the last period of days, where `stock` and `period` are strategy parameters.
- Periodic learns every `period_of_learning` days, where `period_of_learning` is a strategy parameter. He works on other days. For example: for `period_of_learning` = 10, he will learn on turns 10, 20, 30, etc.
- Random works with a probability of  $1 - 1/(\text{day of simulation} + 3)$ , and learns with a probability of  $1/(\text{day of simulation} + 3)$ .

How does a Worker learn?

A Worker learns in one of two ways: either he develops in the current career path or changes it. After changing the career path, the Worker does not forget the level of previous career paths. Although he can only use production bonuses from the current path, if he changes the path to one he had before, he does not start at level 1, but from the one he had before. For example: A Farmer at level 3 became a Miner. He had never been a Miner before, so he becomes a Miner at level 1. Then after some time, he changed his path back to Farmer. Now he is a Farmer at level 3 (instead of 1), because he reached that level before he stopped being a Farmer.

Workers have one of two career change strategies:

- Conservative never changes his career path.
- Revolutionary once every 7 days calculates  $n$  as  $\max(1, \text{his id modulo } 17)$ . Then he chooses the career path that gives a bonus corresponding to the product that appeared most frequently (in

terms of the total number of units in the sales offers of Speculators and Workers) in the last  $n$  days. On other days he does not change his career path. If he chose the career path that he currently has in the above way, he develops in it instead of changing it. For example: on days 1-6 he does not change the path, and on turn 7 he chooses a potentially new career path. Then on turns 8-13 he does not change the path, and on 14 he chooses a potentially new one, etc.

What does a worker produce?

What a worker produces is determined by his production strategy. We have several such strategies:

- Shortsighted always produces the product that had the highest average price the day before (for the definition of average price, see the end of the text).
- Greedy always produces the product that will give him the highest profit this turn. Profit is calculated by multiplying the number of products that we would have produced by the average price of the product from the previous day.
- Averager always produces the product whose maximum average price over the last `average_production_history` days was the highest, where `average_production_history` is a parameter of a given worker.
- Prospective always produces the product whose price increase was the highest over the last `perspective_history` days, where `perspective_history` is a parameter of a given worker. By price increase, we mean the difference between the average price of a product now and `perspective_history` days ago.
- Random always produces a random product.

The number of items produced is determined by the Worker's base productivity vector and the productivity bonus. For example: we produce Diamonds. Base is 100 per day, but we have a 50% bonus, so we produce  $100 + 50\% * 100 = 150$  per day. A given Worker always produces one type of product per day.

What does a worker sell?

A Worker sells everything he produced that day, except for diamonds. If he bought something earlier (or had it from the start) and it was left over, he doesn't sell that.

Please note that this means that a Worker cannot eat food, wear clothes, use tools and computer programs that he himself produced, because the stock market buys all sales offers from Workers at the end of the day.

What does a worker buy?

A Worker can have one of four strategies for buying and using computer programs:

- Technophobe only buys 100 units of food a day.
- Neatnik buys 100 units of food a day and ensures he has at least 100 pieces of clothing for the next round (taking into account the wearing out of clothes at the end of this round).

- Mechanized buys 100 units of food, a number\_tools (where number\_tools is a strategy parameter) tools per day and takes care of clothes just like a Neatnik.

- Gadgeteer buys 100 units of food, number\_tools (where number\_tools is a strategy parameter) tools per day and takes care of clothes just like Neatnik. Additionally, he always buys as many programs as he has produced products in the current round. He uses programs as soon as he can in order from those with the highest level. For example: We have 5 level 2 programs and 3 level 1 programs. We produce 6 tools. We will produce 6 tools: five with a quality level of 2 and one with a production level of 1. At the end of the day, we will have 2 level 1 programs left because we did not use them.

## Career paths

The following career paths are available:

- Farmer
- Miner
- Craftsman
- Engineer
- Programmer

Each path can have any positive level, which gives a production bonus.

## Level bonus

1 +50% of the base value

2 +150% of the base value

3 +300% of the base value

each next level +300% + cumulative +25% for each

level above 3. Percentages refer to the

base value.

The production bonus is applied only to the product related to the given career path:

Farmer Food

Craftsman Clothes

Engineer Tools

Miner Diamonds

Programmer Computer programs

## Products

The following products are available in the simulation:

- Diamonds
- Food
- Clothing
- Tools
- Computer programs

Each product has its unique uses:

Food is essential to every working laborer, and each working laborer consumes 100 units of food per day. If a Laborer did not eat the previous day, they get a productivity bonus of -100%. If they did not eat for two days in a row, they receive a -300% bonus instead. After the third day, the Laborer dies. A dead Laborer does not perform any actions, and their game score is zero diamonds. If the Laborer was learning on a given day, we assume that they ate in the canteen for free and reset the counter of how many days in a row they did not eat.

Clothing (has quality levels) Having fewer than 100 clothes at the beginning of the turn results in the Laborer receiving a productivity bonus equal to  $-\text{penalty\_for\_lack\_of\_clothes}\%$ , where  $\text{penalty\_for\_lack\_of\_clothes}$  is a simulation parameter. Clothes wear out after  $y^2$  turns of use, where  $y$  is the quality level of the clothes.

Tools (have quality levels) Each tool provides a productivity bonus of  $+y\%$  (if we have multiple tools, the bonuses add up), where  $y$  is the tool's quality level. Tools wear out completely after a given turn (in which we produce).

Diamonds are the final currency in the game.

Computer Program (has advancement levels) It allows setting the quality level of tools and clothes to the program's advancement level during production. If the Laborer is a Programmer, the program's advancement level is their career path level, otherwise it's 1. One program works on one tool or piece of clothing. Programs are single-use, so they wear out after their use.

### Description of the Stock Exchange

On the stock exchange, there are buy and sell offers (with price and unit numbers from Speculators and without a price, but with unit numbers from Laborers). You can't sell/buy diamonds on the stock exchange! For computer programs, tools, and clothes, we have separate offers for different levels of program advancement.

The exception is the Laborer's offer to buy a program - they just want to buy a number of programs, tools, clothes, with the highest quality/advancement level available.

On the stock exchange, exchange can take place according to one of three strategies:

- Capitalistic where transactions of Laborers with the most diamonds are first realized,
- Socialist where the order is reversed,
- Balanced where the socialist and capitalist strategy alternate in subsequent turns.

In case of an equal number of diamonds, the Laborer's id decides the order. In other words, we compare pairs of coordinates (number of diamonds, Laborer's id) once in ascending order and once in descending order.

When it's their turn, the Laborer makes a transaction according to the most advantageous of the offers available on the market. The Laborer executes all their orders at once, first all sales and then all purchases. Speculators' and Laborers' offers can be partially realized. The order of selling and buying products for a given Laborer is always the same: Food, Clothing, Tools, Computer Programs. In the case of computer programs, clothes, and tools, Speculators' sales offers are sorted as pairs (advancement/quality level non-increasing, price non-decreasing). This means that a more advantageous offer is, for example, a higher-level program regardless of the price, and for programs of the same level, the price determines the attractiveness of the offer.

Example: a given Laborer wants to sell 5 units of food, and the Speculators' buying offers on the stock exchange are (1 food for 3 diamonds), (100 food for 4 diamonds), and (3 food for 5 diamonds). The Laborer will earn  $5 * 3 + 4 * 2 = 23$  diamonds. The remaining offers on the stock exchange are (1 food for 3 diamonds) and (98 food for 4 diamonds). The same Laborer wants to buy 4 tools (does not state their level). The Speculators' offers are (5 level 3 tools for 29 diamonds) and (100 level 2 tools for 1 diamond). Assuming that the Laborer had 7 diamonds earlier, they have exactly 30 after the sale, so they can only afford 2 tools, one level 3 (as they first buy the best) and one level 2. Therefore, the remaining offers on the stock exchange are (4 level 3 tools for 29 diamonds) and (99 level 2 tools for 1 diamond). Now we proceed to the realization of offers of the next Laborer in order.

If the Laborer fails to sell what they wanted (because the Speculators' buying offers ran out), the bank buys the rest of the products for sale at the lowest buying price for the product from the previous day (turn) or the price from the "zero" turn if there were no buying offers for the product the previous day.

Unrealized purchase and sale offers from Speculators and purchase offers from Laborers disappear and do not carry over to the next turn.

The second type of Agent is the Speculator. Instead of producing, they trade on the stock market. They produce nothing themselves, but have a set budget of diamonds per turn, which they can use to buy items. Any unspent budget is wasted and completely renewed at the beginning of the next turn. Initially, the speculator does not have any items. The description of the stock market is in the next section.

Every speculator, if they place a buy offer, purchases 100 units of a given product (software, clothes, and tools of various levels are considered as different products here). If they place a sell offer, they sell everything they have in one offer. Speculators have different trading strategies (the same for every type of product and level of software advancement, clothing quality, and tool level, from 1 to the turn number of the simulation):

- Average speculators place buy and sell offers 10% below and 10% above the arithmetic mean of the average prices from the last "speculator\_history\_average" (speculator parameter) days. If a given speculator does not have a given product, they only place a buy offer 5% below this value. The average speculator sells everything they have each turn (of course, except for diamonds, see stock market description).
- Convex speculators buy only when the function formed from the average prices of the last 3 days is strictly convex, and sell when it is strictly concave. Otherwise, they do nothing. Buy and sell prices are respectively 10% below and 10% above the average price of the last day.
- The market-regulating speculator does nothing in the first turn. In subsequent turns, they take the average price from the previous day and multiply it by  $p_i / \max(p_i - 1, 1)$ , where  $p_i$  is the number of products of a given type offered for sale by workers in turn  $i$ . Then they place a buy offer 10% below this price and sell 10% above.

To handle writing and reading in the JSON format, you can use the moshi library:  
<https://github.com/square/moshi> or Gson

Additional assumptions and definitions:

- In the case of any looking back  $x$  turns: If less than  $x$  full turns have elapsed since the beginning of the simulation, we just take what is there. At the beginning, the costs of all products (except diamonds) will appear in the "zero" turn. We interpret this as there being one Speculator's buy offer for each product in the "zero" turn (1 product for ... diamonds). The price for different levels of software advancement is the same for all in the "zero" turn. Similarly, the price for different levels of clothing/tool quality. For example, the average from 10 turns needed to make a decision in turn 5 is the average from the first 5 turns (4 real and the artificial zero one), the same goes for counting convex/concave functions. However, if in turn 100 we are counting the average from the last 5 days, we only take the last 5 days, the zero turn doesn't matter then.
- We assume that diamonds are perfectly divisible like bitcoins.
- The IDs of the Workers are pairwise different.
- By the average price of a given day, we mean the weighted average of prices with weights being the number of items. We take into account all the buy and sell offers of Speculators, but only those

realized. If there were no realized offers of a given product, the average price is the price from the "zero" turn of this product. For example, we have offers (5 clothes for 1 diamond) and (2 clothes for 3 diamonds). The average price of clothes that day is  $(5 * 1 + 2 * 3) / 7$ .

- Items owned by agents at the start of the simulation (given at the input) are treated as level 1 items (for clothes, tools, and programs) and as purchased. This means that we do not sell them, but use them.

## FAQ

- Does the Worker use food/tools/clothes/software in the turn when they learn? Answer: No.

- What happens to the unused software? For example, I produce 10 tools, but I have 100 software? Answer: At the end of the turn, the Worker loses 10 software (because they used so many to produce tools), but 90 remains and they can use it in the following turns.

- What exactly does the use of an item mean? Answer: It means that such an item should be removed from the simulation at the moment of its use. There are no items with a "used" status, all such go to the "bin".

- What to do when choosing a product/career path, in case of a tie (according to the established measure)? Answer: We resolve ties taking into account the order of career paths/products in the table in the text. For example, if we want to produce and according to our strategy Tools and Software are equally good, we produce Software, because they are further in the table. Similarly, if we want to change the career path and Farmer and Craftsman are equally good, we choose the latter in the table, i.e. Craftsman.