CS 3050: Group Project

April 12, 2016

DUE: 05/05/2016

Topic: Optimal trading strategies

This is the final project of the course. The project can be done alone or in a group of at most 3 people. The project contributes to 12% of your total grade. You should send an email to Eric, Elizabeth and myself by April 21 with the subject "CS3050 project" indicating whether you are choosing to do the project alone or in a group. If you choose to do the project in a group, please include the name of your teammates in the project (one email per team is fine).

Description

The purpose of this project is to build upon your algorithm design and general programming skills. Algorithms play an important role in trading strategies at investment firms. In this project, you will model and implement a relatively "simple" trading strategy. Imagine you are working at an investment firm, say X, and are responsible for trading a stock for a company, say Y. The company has an excellent predictive modeling team and can predict the stocks for the company Y for the next n days with reasonable accuracy. Let the predicted prices be $p_1, p_2, \ldots, p_n$. So p_1 is the predicted stock price at day 1, p_2 is the predicted stock price at day 2, and so on. Your goal is to buy and sell stocks of Y in order to maximize profits. For this you shall be using r-trades strategy, described below.

An r-trades strategy, for an integer r, is a sequence of $2k$ integers :

$$\text{Strat} = b_1, s_1, b_2, s_2, \ldots, b_k, s_k \text{ where } k \leq r.$$

1. The sequence Strat means that you will make k-trades as follows.
   - You will buy Y's stock at day b_1
   - You will sell Y's stock at day s_1
   - You will buy Y's stock again at day b_2
   - You will sell Y's stock again at day s_2
   - ...
2. The sequence Strat is strictly increasing, *i.e.*, $b_1 < s_1 < b_2 < s_2 < \ldots b_k < s_k$.
3. The numbers in Strat are at least 1 and atmost n.

Your task is to write a program that given n, r and $p_1, \ldots, p_n$, outputs the r-trades strategy that maximizes the profit, *i.e.*, maximizes the quantity

$$\text{Profit(Strat)} \;=\; (p_{s_1} - p_{b_1}) + (p_{s_2} - p_{b_2}) + \ldots (p_{s_k} - p_{b_k}).$$

For example, suppose $n = 6$ and the predicted prices for the next 6 days are:

$$p_1 = 1, p_2 = 3, p_3 = 4, p_4 = 6, p_5 = 2, p_6 = 10.$$

If $r = 1$ then the optimal strategy is $b_1 = 1, s_1 = 6$, *i.e*, buy at day 1 and sell at day 6. If $r = 2$ then the optimal strategy is $b_1 = 1, s_1 = 4, b_2 = 5, s_2 = 6$, *i.e*, buy at day 1, sell at day 4, buy at day 5 and sell at day 6. If $r = 3$ then the optimal strategy is again $b_1 = 1, s_1 = 4, b_2 = 5, s_2 = 6$.

Constraints

The constraints of the program include:

1. The program can be written in C or C++. If you want to use a different language, please check with us.
2. Either use a netbeans project or use a Makefile to compile and run the program.
3. Include a README file to tell the TA's how to run your program.
4. You should include a project report in which you detail your efforts, the algorithms used (along with their complexity) and the contribution of your team members. We shall use this report primarily to give you partial credit if your program does not work.
5. A moderate amount of error checking and resource management is required. Your application should check that each line from the input file is properly formatted, that the file is successfully opened, the file is successfully closed upon reading of the file and that all allocated space is de-allocated at the exit. If the input is not properly formatted, you should report the error on stdout and exit the program.
6. You may use standard input/output libraries, arrays, vectors, stacks, heaps, lists and queues. If using other libraries, please please check with us.
7. A moderate amount of formatting and documentation is required. Comments should be descriptive and used to illustrate the purpose and inner workings of an algorithm or function; they should not be used to annotate each line or self-evident logic.
8. For substantial credit, your program should work for at least the cases when $r = 1, r = 2$ and $r = 3$.

Input

Your program should take at least three arguments on the command line. The first argument is the number of days n, the second is the parameter r and the third argument in an input file which contains the list of predicted prices $p_1, p_2, \ldots p_n$. Each line of an input file consists of a non-negative integer followed by a newline character. Each integer is a string of digits. The ith line of this file is p_i, the predicted price of stock at day i. A sample input file is included on Blackboard.

Output

Each line printed to standard output should consist of a non-negative integer followed by a newline character. The first line of the output is value of b_1, the second line of the file is s_1, the third line of the file is b_2, the fourth line of the file is s_2 and so on. A sample output file is included on Blackboard which includes the optimal strategy for the prices in the sample input file for $r = 3$.

Due Dates

- April 21 is the date that you must inform us about the composition of your team.
- During the week of April 25-April 29, please try to meet one of the TAs or myself during office hours and show the progress of the project. If the times do not work, please send us an email and we will try to find additional time to check the progress.
- The final submission is due May 5 at 11:59:59 pm.

Submission

You should place all your submission material into a folder named as follows: *pawprint1_pawprint2_pawprint3_project*. Where the *pawprint1_pawprint2_pawprint3* is the list of you and your group member pawprints.
You will submit one file, a compressed directory containing all the source code and appropriate Makefile(s). The naming convention should be: *pawprint1_pawprint2_pawprint3_project.tgz* or *pawprint1_pawprint2_pawprint3_project.zip*.

Grading

There are 120 points possible for this assignment. The grade breakdown is as follows:

- 15 points for README, error checking and resource management.
- 15 points for general programming style and adherence to the constraints.
- 20 points for the project report.
- 10 points for correctly inputting the predicted prices.
- 60 points for correctly outputting the strategy.