Linear Algebra Assignment - I

TA: Anirban, Mayank, Amal

August 29, 2018

Instructions

- This is a coding assignment. The code has to be written in Python. Please use Python version 2.7 to solve all the problems.
- The assignment deliverable is your python code and your report of observations.
- Send the files in a zip file to **E0226@outlook.com** with the following format for both zip file name and subject name of email: LA_1_< last 5 digits of serial number >.zip.
- You are encouraged to discuss among yourselves, but DO NOT COPY. It can be fatal.
- Read the complete assignment carefully, before attempting it.
- Follow the instructions on how to run the code, format for input and output file and naming conventions. — Detailed Instructions
- The submission deadline is **September 20, 2018**.

Best of luck for the assignment. HAPPY CODING!

Problem - I
Can you prepare the potion - LinAlgebrica?

First Part (1.5 Points)

Potions is a core class and subject taught at Hogwarts School of Witchcraft and Wizardry. In this class, students learn the correct way to brew potions. They follow specific recipes and use various magical ingredients to create the potions, starting with simple ones and moving to more advanced ones as they progress in knowledge.

In their last potion class before Christmas, Professor Snape who is the Potion Master(who tended to favour students from Slytherin House more than other

students), gave a task to four first year students from the House of Gryffindor: Harry, Ron, Hermione and Neville Longbottom. He asked them to make a potion - ***LinAlgebrica***. Each of them will have to prepare in different amounts as specified by Prof. Snape (say, c_1, c_2, c_3, c_4 respectively). To prepare the potion, four ingredients are required - 1. *Moondew* 2. *Cauliflower seeds* 3.*Flobberworm Mucus* 4.*Dragon blood*. The quantity for each of these ingredients has a maximum limit available in the dungeon-inventory (say, these constant quantities are - m_1, m_2, m_3, m_4 respectively).

Professor Snape has specified each student a fixed percentage of each of the ingredients (p_{ij} for i^{th} student and j^{th} ingredient, $i \in \{1, 2, 3, 4\}$ and $j \in \{1, 2, 3, 4\}$) they can use. Total quantity of each ingredient should not exceed what is available in inventory (upperbounded by m_i, $i = \{1, 2, 3, 4\}$). Harry, Ron, Neville and Hermione - all four of them being novices in algebra, and you being an expert in solving mathematical equations, seek your help! Your task is to make their life easier by telling them whether they will be able to make *LinAlgebrica*. If so, what is the quantity of each ingredient required.

Input format - All the inputs will be given in text file in the following format: The first line contains c_1, c_2, c_3 and c_4 in space separated format. Next 4 lines will contain the percentage of each ingredient (p_{ij}) in space separated order. Last line will contain maximum amount of each ingredient available in inventory i.e. m_1, m_2, m_3, m_4. The order of ingredients is maintained.

Output Format - If no such solution exists, print **NOT POSSIBLE, SNAPE IS WICKED!**. If there is exactly one solution, print **EXACTLY ONE!** in the first line and in the next line print the space-separated solution values in order. If multiple solutions exist, print **MORE THAN ONE!** and report one feasible solution in the second line and the generalized solution in the third line. The way you want to mention the generalized solution is up to you.

Second Part (1.5 Points)

To make their life more miserable, Prof. Snape introduced more students from Gryffindor house while simultaneously adding more ingredients. The total number of students is now n and there are k types of ingredients. Your job is now a bit difficult! The task though is still the same - find out whether they will be able to make the potion or not! And if the answer is yes, mention the exact amount of each ingredient required.

Input Format - First line of input contains values of n and k in order. Next line contains n integers. Next n lines contain the percentage of ingredients. The last line contains k integers. You can understand that the format is similar to the previous problem, except the first line.

Output Format - Same as mentioned in the output format section in part one.

Bonus Question (1 Point)

If it is not possible to prepare the potion (i.e. no possible solution) because of limitation in ingredient quantity, can you tell the maximum amount of potion that can be prepared out of the given ingredients?

Problem - II
Search for Philosopher's Stone - More obstacles!

First Part (2 Points)

In quest of the Philosopher's Stone hidden behind the trapdoor, Harry, Ron and Hermione guessed that due to the sudden rise in application of Linear Algebra in the wizard world, they might be facing a series of obstacles which may involve some maths. So, you being a math wizard and a bosom friend of theirs, they form a team with you. It turns out that their guess was right. The very first puzzle involves solving a mathematical problem as follows : There is a huge squared checkerboard of size n by n. Each and every box in that board contains a real number. n can be arbitrarily large.

You have three acceptable operations at your disposal - (I) *switch two rows*, (II) *multiply a row with a non-zero constant* and (III) *add a row with another row multiplied by some constant*. What you have to do is modify the checker board (read matrix) in such a way that the product of initial matrix and final matrix becomes an identity matrix.

If such a matrix is not possible, then you have to speak out "Not possible" to get through the step.

Input format : First line contains the value of n. Next n lines contain the value of the square matrix in row-major order.

Output Format - If you are unable to find such a matrix, print **ALAS! DIDN'T FIND ONE!** and report the row operations in each line. For example, if your first operation is to switch two rows, print **SWITCH** followed by indices of two rows in space separated format. Print **MULTIPLY** and the multiplicative constant and row index in space separated way for the second operation. For the third operation, print **MULTIPLY&ADD** followed by the multiplicative constant, index of the multiplied row and index of the current row in space separated format. If such a matrix exists, print **YAAY! FOUND ONE !** and then report the final matrix in row major form, each line will correspond to a row of your final matrix. Please report the row operations in the subsequent lines in the same format mentioned for the previous case. Keep in mind that row index starts from 1. — See example format

Second Part (0.5 Point)

You will have to report the time taken for the process to complete. Compare it with the standard library API available in Python and mention the difference. Can you find a way to improve the efficiency of your algorithm in terms of time taken?

Third Part (0.5 Point)

Consider you are given the same three operations, but this time instead of row operations you are allowed to do column operations. Can you find an inverse? Explain why or why not.

Bonus Question (1 Point)

Say you have only two out of the three operations at your disposal. Given a matrix, is it possible to get identity matrix? Also, what can you say about the complexity of the algorithm?

How to run

You need to write your code in two separate python files for two problems - *first_problem.py* and *second_problem.py* You should print your output in terminal as well as into three files for the two problems. The naming convention for the output files is mentioned below.

Command to run the first problem: If I want to check the first part of the first problem, I should be able to run it using the following command: *python first_problem.py –part=one <input_file_path>*

To check the second part of the first problem, I should be able to run it using the following command: *python first_problem.py –part=two <input_file_path>*

Command to run the second problem: I should be able to run it using the following command: *python second_problem.py <input_file_path>*

File naming convention

- The name of output file for problem one: first part needs to be *output_problem1_part1.txt*
- The name of output file for problem one: second part needs to be *output_problem1_part2.txt*
- The name of output file for problem two needs to be *output_problem2.txt*
- Prepare a report and name it *LA_Report_1.pdf*. It should contain the solutions for both the bonus questions and second, third part of the second question.

Sample Input and Output

** Please look into the input and output files provided along with for exact format.

First Problem

First part -

Input

27 21 22 16
0.1 0.3 0.2 0.4
0.3 0.3 0.2 0.1
0.4 0.1 0.4 0.1
0.1 0.2 0.1 0.2
33 35 46 57

Explanation of input

The first line contains four integers denoting the amount of potion each person has to prepare. Each of the next 4 lines contain 4 real numbers. Each line signifies the percentage of each ingredient that person can use, e.g the second line's four values indicate what percentages of each ingredient the first person can take. The next three lines are populated in similar fashion. The order of ingredients is maintained. The last line contains four integers - denoting the maximum amount of ingredients available at the dungeons' inventory(order of ingredients is same as before - first integer is the first ingredient's quantity, second is second ingredient's and so on).

Output

EXACTLY ONE!
30 20 10 40

Second part -

Input

3 2
10 20 30
0.1 0.2
0.3 0.4
0.2 0.2
100 100

Output

NOT POSSIBLE, SNAPE IS WICKED!

Second Problem

Input

2
10 0
0 10

Output

YAAY! FOUND ONE!
0.1 0
0 0.1
MULTIPLY 0.01 1
MULTIPLY 0.01 2

Input

4
1 2 3 4
1 2 4 5
2 4 6 8
9 2 1 4

Output

ALAS! DIDN'T FIND ONE!
SWITCH 3 4
MULTIPLY&ADD -1 1 2
MULTIPLY&ADD -9 1 3
MULTIPLY&ADD -2 1 4

Explanation of output

The second line of the previous output indicates switching between third and fourth row. The next line denotes first row is multiplied by -1 and then added to second row. The fourth line indicates the third is modified in the following manner: first row is multiplied by -9 and then added to third row.