Assignment 2

CS221: Data Structures and Algorithms

Usman Institute of Technology

Fall 2019

Release: 6 November 2019 11:00 PM
Submission: 12 November 2019 08:30 AM
Submission Mode: Soft-copy

- How to submit:
  - Create an account on http://www.turnitin.com/ as a Student
  - Use following information at time of sign-up

    CS Section A
    - Class ID: 22664649
    - Enrollment Key: DSFALL19CSA

    CS Section B
    - Class ID: 22664651
    - Enrollment Key: DSFALL19CSB
- You have to submit single .py file
- Make sure that function names must be similar as asked in the assignment.
- **You can't use the built-in function of numpy package.**
- You must read Academic Integrity at the end of this document.

1. Write a class **MatrixAssignment** and following functions in the class. I would encourage to use numpy library for matrix handling.

   a. [15 points] Write a function **ReadFromFile** which takes a parameter "filename" to read the matrix from the file and returns a matrix (numpy object).

```
          def ReadFromFile(filename:string)
```

The format of file is given below:
The first line of the file contains two integer R and C, separated by space. The next R lines contain C integers separated by space
Example 1:

```
3 3
-1 0 2
2 3 11
14 2 3
```

Example 2:

```
2 3
-1 0 2
2 3 11
```

   b. [15 points] Write a function Multiply() which takes two matrices matrix1 and matrix2, both are objects of numpy.ndarray, as parameters and returns a new matrix. The new matrix is the multiplication of two matrices.

```
          def Multiply(matrix1:numpy.ndarray, matrix2: numpy.ndarray)
```

c. [15 points] Write a function Inverse which takes a parameter **matrix** and returns a matrix which is the inverse of the given array. If the inverse of matrix is not possible then the function should raise an exception.

```
          def Inverse(matrix:numpy.ndarray)
```

For matrix inversion, you can use Elementary Row Operations. You can use the following link to learn about this. The objective of this assignment is not to learn this topic as we believe that this is already covered in your Linear Algebra course.

https://www.mathsisfun.com/algebra/matrix-inverse-row-operations-gauss-jordan.html

d. [15 points] Write a function **SolveSystemOfEquation** which takes two parameters **A** and **B**, both are matrices, and returns a matrix which is the solution of linear equation.

```
          def SolveSystemOfEquation (A:numpy.ndarray, B:numpy.ndarray)
```

We know that a linear system of AX = B can be solved by finding X = BA^{-1}. So, it can be solved by multiplying matrix B with inverse of matrix A. If matrix cannot be multiplied or inverse cannot be found, then function should raise an exception.

Representing System of Linear Equations:

The following is a set of equations for a linear system:

$$\begin{gathered} x_1 + 2x_2 + 3x_3 = 5 \\ 2x_1 + 5x_2 + 3x_3 = 3 \\ x_1 + 2x_2 + \ 8x_3 = 17 \end{gathered}$$

We know, in a Matrix form this system can be written as Ax = b

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 2 & 8 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, b = \begin{bmatrix} 5 \\ 3 \\ 17 \end{bmatrix}$$

If A and b are stored in separate files, such that matrix_a.txt and matrix_b.txt, then content of both files will look like the following:

```
matrix_a.txt
3 3
1 2 3
2 5 3
1 2 8
```

```
matrix_b.txt
3 1
5
3
17
```

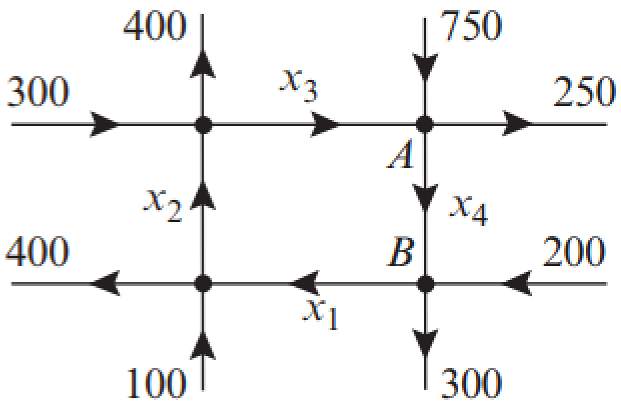
If we have implemented the functions discussed in the Question 1, then we can read these matrices by using the following code:

```
ma = MatrixAssignment()
A = ma.ReadFromFile(“matrix_a.txt”)
b = ma.ReadFromFile(“matrix_b.txt”)
```

Even we can solve this system of equation by calling the following line:

```
solution = ma.SolveSystemOfEquation(A, b)
```

2. [40 points] The following figure shows a network of one-way streets with traffic flowing in the directions indicated. The flow rates along the streets are measured as the average number of vehicles per hour.
    a. Set up a linear system who solution provides the unknown flow rates. Store A and b matrices of the system in separate text files "**matrix_a.txt**" and "**matrix_b.txt**". [20 points]
    b. Write a function **SolveNetworkFlow** (this function is not part of class) which returns a matrix. The return matrix is the solution of the system of equations. [20 points]



Academic Integrity

Each student in this course is expected to make sure that any work submitted by a student in this course for academic credit will be the **student's own work**. Scholastic dishonesty shall be considered a serious violation of these rules and regulations and is subject to strict disciplinary action. Scholastic dishonesty includes, but is not limited to, cheating on exams, plagiarism on assignments, and collusion.

PLAGIARISM: Plagiarism is the act of taking the work created by another person or entity and presenting it as one's own for the purpose of personal gain or of obtaining academic credit. Plagiarism includes the submission of or incorporation of the work of others without acknowledging its provenance or giving due credit according to established academic practices. This includes the submission of material that has been appropriated, bought, received as a gift, downloaded, or obtained by any other means. Students must not, unless they have been granted permission from all faculty members concerned, submit the same assignment or project for academic credit for different courses.

CHEATING: The term cheating shall refer to the use of or obtaining of unauthorized information in order to obtain personal benefit or academic credit.

COLLUSION: Collusion is the act of providing unauthorized assistance to one or more person or of not taking the appropriate precautions against doing so. Any student caught violating academic integrity will suffer an academic penalty. All violations of academic integrity will also be immediately reported to the Disciplinary Committee. Any student violating academic integrity a second time in this course will receive a failing grade for the course, and additional disciplinary sanctions may be administered through the Disciplinary Committee.

Conclusively, each student need to be take care of:

1. You must not share your solutions with other students. You are encouraged to discuss the problems but each student is supposed to take care of his or her own solution.
2. You must not submit solution of other students as yours.
3. You must duly cite all resources you used in development of your solution.