12.248

Harsha-EE25BTECH11026

September 17,2025

Question

A square matrix **A** will be lower triangular matrix if and only if a_{MN} represents an element in the M^{th} row and N^{th} column of the matrix

1. $a_{MN} = 0, N > M$
2. $a_{MN} = 0, M > N$
3. $a_{MN} \neq 0, M > N$
4. $a_{MN} \neq 0, N > M$

Theoretical Solution

A lower triangular matrix of size $m \times n$ is defined as for any element a_{ij} in the matrix,

$$a_{ij} = 0 \; \forall \; i < j \tag{1}$$

By the definition, option (1) are correct.

C Code -Checking whether the matrix is lower triangular matrix

```c
#include<stdio.h>

void check_ltm(int m, int n, double matrix[m][n]){
        if(m!=n){
                printf("It is not a lower triangular matrix\n");
                return;
        }
        for(int i=0;i<m;i++){
                for(int j=0;j<n;j++){if(j>i && matrix[i][j]!=0){
                        printf("It is not a lower triangular matrix\n");
                        return;
                        }
                }
        }
        printf("It is a lower triangular matrix\n");
}
```

Python+C code

```python
import ctypes
import numpy as np

lib=ctypes.CDLL("./libltm.so")

lib.check_ltm.argtypes=(ctypes.c_int , ctypes.c_int, np.ctypeslib
    .ndpointer(dtype=np.float64, ndim=2 , flags="C_CONTIGUOUS"))

lib.check_ltm.restype= None

#Example
A=np.matrix([[2,0,0],[3,1,0],[8,7,6]]).astype(np.float64)

lib.check_ltm(A.shape[0],A.shape[1],A)
```

Python code

```python
import numpy as np

def check_ltm(matrix):
    m,n= np.shape(matrix)
    if(m!=n):
        print("It is not a lower triangular matrix")
        return
    if np.all(matrix[np.triu_indices(m,k=1)]==0):
        print("It is a lower triangular matrix")
    else:
        print("It is not a lower triangular matrix")

def generate_ltm(n, low=0 , high=10 ):
    A= np.random.randint( low , high , size=(n,n))
    return np.tril(A)

L=generate_ltm(4)
print(L)
```