

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

-----oOo-----



TOÁN ỨNG DỤNG THỐNG KÊ

Lab 2

Họ tên : Lê Nguyên Bình Nam
MSSV: 20127567

1. Tìm ma trận nghịch đảo:

- Ý tưởng: Xây dựng một ma trận đơn vị I. Cùng với ma trận ban đầu là A. Với mỗi bước thực hiện để đưa ma trận A về ma trận đơn vị, ta thực hiện tương tự với ma trận đơn vị I ban đầu. Kết quả sau cùng ta thu được ma trận nghịch đảo của A.
- Cách thực hiện & mô tả hàm:
 - Create Identity Matrix: Hàm khởi tạo ma trận đơn vị.

```
def create_identity_matrix(n):  
    return [[1 if i == j else 0 for j in range(n)] for i in range(n)]
```

[13] ✓ 0.3s

- Swap_row: Hoán đổi vị trí 2 dòng trong cùng 2 ma trận.

```
def swap_row(A, I, i, j):  
    col = len(A[0])  
    for col1 in range(col):  
        temp = A[i][col1]  
        A[i][col1] = A[j][col1]  
        A[j][col1] = temp  
    for col1 in range(col):  
        temp = I[i][col1]  
        I[i][col1] = I[j][col1]  
        I[j][col1] = temp  
    return A
```

[14] ✓ 0.4s

- findNewRow: nhận vào ma trận A, I khi tại vị trí $A[i][j] = 0$, tìm hàng mới để thay thế cho giá trị khác 0. Trả về 1 nếu như thay thế được, 0 thì ngược lại.

```
def findNewRow(A, I, i, j):  
    row = len(A)  
    col = len(A[0])  
    for x in range(i+1, row):  
        if A[x][j] != 0:  
            swap_row(A, I, i, x)  
            return 1  
    return 0
```

[15] ✓ 0.3s

- sumRow, divRow: các phép biến đổi sơ cấp trên dòng (chia 1 dòng cho 1 số và tổng của hàng này với k lần hàng khác)

```
def sumRow(A,I,i,j,k):  
    n = len(A)  
    for x in range(n):  
        A[i][x]=A[i][x]+k*A[j][x]  
        I[i][x]=I[i][x]+k*I[j][x]  
  
def divRow(A,I,i,k):  
    n = len(A)  
    for j in range(n):  
        A[i][j]=A[i][j]/k  
        I[i][j]=I[i][j]/k
```

[16] ✓ 0.3s

- inverse(A)
 - Kiểm tra ma trận có phải là ma trận vuông hay không
 - Kiểm tra phần tử đầu tiên có khác 0 hay không, nếu có thì thay thế bằng hàng khác có phần tử đầu hàng khác 0.
 - Xây dựng ma trận đơn vị
 - Với mỗi hàng, ta xét nếu phần tử đó khác 0 thì sẽ thực hiện thuật toán Gauss-Jordan, nếu phần tử đó bằng 0, ta thực hiện tìm kiếm dòng thay thế. Nếu như không tồn tại dòng thay thế, ta kết luận ma trận không khả nghịch

```

def inverse(A):
    if (len(A) != len(A[0])):
        print("Ma tran khong kha nghich")
        return None

    n = len(A)

    I = create_identity_matrix(n)
    if A[0][0] == 0:
        for i in range(1, n):
            if A[i][0] != 0:
                swap_row(A, I, 0, i)
                break
    pivot = 0
    i=0
    while i < n:
        if A[i][pivot]==0:
            flag = findNewRow(A,I,i+1,pivot)
            if flag == 1:
                continue
            else:
                print("Ma tran khong kha nghich")
                return None
        else:
            divRow(A,I,i,a[i][pivot])
            for j in range(n):
                if i!=j:
                    sumRow(A,I,j,i,-a[j][pivot])
            pivot = pivot + 1
            i+=1

    return I

```