

**BÀI BÁO CÁO VIẾT CHƯƠNG TRÌNH GIẢI HỆ PHƯƠNG TRÌNH BẰNG
THUẬT TOÁN GAUSS**



TÊN: BÙI HỒNG DƯƠNG

MSSV:20120273

MÔN: TOÁN ỨNG DỤNG VÀ THỐNG KÊ

LỚP: 20_1 N1

I Ý TƯỞNG THỰC HIỆN:

Dựa vào thuật toán gauss để giải 1 hệ phương trình n ẩn:

- + Sắp xếp các dòng sao cho phần tử khác không đầu tiên bên trái của hàng trên luôn nằm cùng cột hoặc nằm ở phía bên tay trái so với phần tử khác không đầu tiên của hàng dưới.
- + Chuyển leading point của hàng đang xét về 1
- + Thực hiện các phép biến đổi sơ cấp trên dòng để đưa về dạng bậc thang với 1 hàng, sau đó xét tiếp hàng dưới và lặp lại đến hết.

II GIẢI THÍCH HÀM:

```
3
4  def doi_dong(A,B):
5      l=len(A)
6      for i in range (l):
7          A[i],B[i]=B[i],A[i]
8
```

Hàm này để đổi các dòng với nhau

```
def chuyenhesove1(A,j):
    l=len(A)
    chia=A[j]
    if(chia!=0):
        for i in range(1):
            A[i]=A[i]/chia
```

Hàm này để đưa leading point về 1

```
def congvoimatrankhac(A,B,n):
    l=len(A)
    for i in range (l):
        B[i]+=A[i]*n
```

Hàm này dùng để cộng ma trận này với n *ma trận khác.

```

def Gauss_elimination(A):
    m=len(A)
    n=len(A[0])

    for i in range(m):
        dieukien=0
        for j in range(i,m):
            if(A[j][i]!=0):
                dieukien=1
                doi_dong(A[i],A[j])
                break
        if(dieukien==0):
            continue
        chuyenhesove1(A[i],i)
        for j in range(i+1,m):
            bientam=-(A[j][i]/A[i][i])
            congvoimatrankhac(A[i],A[j],bientam)
    print("Ma tran bac thang: ")
    for i in range(len(A)):
        print(A[i])

```

Hàm Gauss này dùng để đưa ma trận ban đầu về dạng bậc thang. Đầu tiên ta sẽ xét xem có hàng nào có leading point đang xét có khác không hay không, nếu có thì ta sẽ đổi nó với dòng đang xét, nếu không thì ta sẽ xét tiếp cột tiếp theo vì tất cả phần tử cột này đều bằng 0 rồi. Sau khi đổi với dòng đang xét, ta sẽ chuyển leading point về 1 bằng cách gọi hàm chuyenhesove1. Rồi ta cho chạy các hàng tiếp theo để biến đổi các hệ số cùng cột nằm dưới leading point về 0 để tạo thành dạng bậc thang (Lấy dòng $A[j]-A[i]*n$ với $n = A[j][i]/A[i][i]$).

```

40     print(A[i])
41 def back_substitution(A):
42     m=Len(A)
43     n=Len(A[0])
44     for i in range(m):
45         if A[i][i]==0:
46             if(A[i][n-1]!=0):
47                 print("VN")
48                 return
49             elif ((m-n+1) <=0):
50                 print("VSN")
51                 return
52     x=[0]
53     for i in range (n-2,-1,-1): #Chạy từ n-1 đến 0 (số phần tử nghiệm)
54         k=0
55         b=A[i][n-1]
56         for j in range(i+1,n-1):
57             b-=(A[i][j]*x[k])
58
59         k=k+1
60         x.insert(0,b)
61     x.pop()
62     print("Nghiem:")
63     for i in range(len(x)):
64         print("x",i,"=",x[i])
65
66
67
68

```

Hàm này dùng để in các nghiệm của hệ phương trình ra. Ta sẽ kiểm tra xem nếu có phần tử $A[i][i]$ nào bằng không mà hệ số cuối của nó khác không hay không. (Vì nếu $A[i][i]$ bằng 0 thì các số nằm bên trái nó đều bằng 0 do ta đã sắp xếp ở trên). Nếu phần tử cuối khác 0 thì nó vô nghiệm, còn nếu nó bằng 0 thì có vô số nghiệm (với điều kiện số dòng \leq số cột). Sau đó ta lưu nghiệm vào list x, tính nghiệm từ dòng cuối lên trên bằng cách thực hiện tính toán theo biểu thức mỗi dòng, mỗi nghiệm tính được ta sẽ thêm nó vào đầu của list x rồi in ra.