

-----  
Nesta aula: Propriedades das operações:  
associatividade, comutatividade, existência de elemento neutro...  
grupos, monóides  
-----

- Uma operação num universo finito  $U=\{0,1,\dots,n-1\}$  pode ser representada por uma lista de listas. Por exemplo, soma modulo 3

```
tabela = [[0,1,2],  
          [1,2,0],  
          [2,0,1]]
```

x+y: linha x, coluna y  
2+1 -> 0

Escreva funções para

- a) `elem_neutro(tabela)` -> calcula o (um) elemento neutro (0 no exemplo) ou -1 se não existe  
Sug. Use uma função auxiliar  
`neutro(x,tabela)` -> x é elemento neutro?
- b) `inversos(tabela)` -> {False,True} (True no exemplo)  
True sse verifica as 2 condições seguintes  
1) existe elemento neutro, seja e  
2) para cada elemento x de U existe um elemento x' de U (o inverso de x) tal que  
 $x+x' = x'+x = e$
- c) `associativa(tabela)` -> {False,True} (True no exemplo)
- d) `comutativa(tabela)` -> {False,True} (True no exemplo)
- e) `monoide(tabela)` -> {False,True} (True no exemplo)
- f) `grupo(tabela)` -> {False,True} (True no exemplo)
- g) Indique um exemplo de um grupo com poucos elementos que não seja comutativo. Verifique, usando as funções que escreveu.
- h) Indique um exemplo de uma operação definida num universo com poucos elementos que não seja associativa.
- i) A eficiencia da função que escreveu nas alinea d) pode ser melhorada. Reescreva a função de modo a haver menos testes de igualdade.

---

Notas.

Seja + uma operação definida num universo U.

- Diz-se que existe elemento neutro se existe um elemento e pertencente a U tal que para todo o x de U se verifica  $e+x=x+e=x$
  - Diz-se que + é associativa se para quaisquer elementos x,y,z de U se verifica  $(x+y)+z = z+(y+z)$
  - Diz-se que + é comutativa se para quaisquer elementos x,y de U se verifica  $x+y = y+x$
  
  - (U,+) é um monoide se a operação é associativa e existe elemento neutro.
  
  - (U,+) é um grupo se a operação é associativa, existe elemento neutro e cada elemento tem um inverso.
- 

```
#-- a) -----
def elem_neutro(t):
    "Elemento neutro de t ou -1 se nao existe"
    n=len(t)
    for x in range(n):
        if neutro(x,t):
            return x
    return -1

def neutro(x,t):
    " x é um elemento neutro de t?"
    n=len(t)
    for z in range(n):
        if t[x][z] != z or t[z][x] != z:
            return False
    return True
```

```

#-- b) -----
def inversos(t):
    "Todos os elementos tem inverso?"
    e=elem_neutro(t)
    if e==-1:
        return False
    n=len(t)
    for x in range(n):
        if not tem_inverso(x,e,t):
            return False
    return True

def tem_inverso(a,e,t):
    "a tem inverso?"
    n=len(t)
    for x in range(n):
        if t[a][x]==e and t[x][a]==e:
            return True
    return False

#-- c) -----
def associativa(t):
    n=len(t)
    for x in range(n):
        for y in range(n):
            for z in range(n):
                xy=t[x][y]
                yz=t[y][z]
                if t[xy][z]!=t[x][yz]:
                    return False
    return True

#-- d) -----
def comutativa(t):
    n=len(t)
    for x in range(n):
        for y in range(n):
            if t[x][y]!=t[y][x]:

```

```

        return False
    return True

#-- e) -----
def monoide(t):
    return elem_neutro(t)!=-1 and asociativa(t)

#-- f) -----
def grupo(t):
    return monoide(t) and inversos(t)

#-- f) -----
def comutativa(t):
    n=len(t)
    for x in range(n):
        for y in range(x+1,n):
            if t[x][y]!=t[y][x]:
                return False
    return True

def search():
    t = [[0,1,2],
         [1,0,0],
         [2,0,0]]
    for v11 in [0,1,2]:
        t[1][1]=v11
        for v12 in [0,1,2]:
            t[1][2]=v12
            for v21 in [0,1,2]:
                t[2][1]=v21
                for v22 in [0,1,2]:
                    t[2][2]=v22
                    if inversos(t) and asociativa(t):
                        print t

```

```
#-----  
t1 = [[0,1,2],  
      [1,2,0],  
      [2,0,1]]  
  
t2 = [[0,1,2],  
      [1,2,1],  
      [2,0,1]]  
  
for t in [t1,t2]:  
    print  
    print "Elem. neutro: ", elem_neutro(t)  
    print "Inversos:      ", inversos(t)  
    print "Asociativa:    ", asociativa(t)  
    print "Comutativa:    ", comutativa(t)  
    print "Monoide:       ", monoide(t)  
    print "Grupo:         ", grupo(t)
```