

## União

### Código:

```
conjuntoA = [1,2,3,4,5]
conjuntoB = [4,5,6,7,8]
conjuntoUniao = []
for elementoDeA in conjuntoA:
    conjuntoUniao.append(elementoDeA)
for elementoDeB in conjuntoB:
    if(elementoDeB not in conjuntoUniao):
        conjuntoUniao.append(elementoDeB)
print(conjuntoUniao)
```

### Pseudo-Código:

#### Algoritmo 1: União(A,B)

Entrada: A e B são conjuntos

Saída: O conjunto resultante da união entre os conjuntos A e B

```
1:  $C_U \leftarrow \{\}$ 
2: para cada  $a \in A$  faça:
3: | Insere( $C_U$ , a)
4: fim
5: para cada  $b \in B$  faça:
6: | se  $b \notin C_U$  então:
7: || Insere( $C_U$ , b)
8: fim
9: retorne  $C_U$ 
```

#### Algoritmo 2: Insere(A,a)

Entrada: A é um conjunto; a é um valor que será inserido em A

```
1: se  $a \notin A$  então:
2: |  $N \leftarrow |A|$ 
3: |  $A_{n+1} \leftarrow a$ 
4: fim
```

### Explicação Código:

O algoritmo 1 realiza a união de dois conjuntos, enquanto o algoritmo 2 realiza uma função de união usada no algoritmo 2.

## Intersecção

### Código:

```
1.conjuntoA = [1,2,3,4,5]
```

```

2.conjuntoB = [4,5,6,7,8]
3.conjuntoIntersecao = []
4.for elementoDeA in conjuntoA:
5.if elementoDeA in conjuntoB:
6.conjuntoIntersecao.append(elementoDeA)
7.print(conjuntoIntersecao)

```

Pseudo-Código:

Algoritmo 1: Intersecção(A, B)

Entrada: A e B são conjuntos

Saída: O conjunto resultante da intersecção entre os conjuntos A e B

1:  $CI \leftarrow \{\}$

2: para cada  $a \in A$  faça:

3: | se  $a \in B$  então:

4: || Insere(CI,a)

5: fim

6: retorne CI

Explicação Código:

O algoritmo 1 realiza a Intersecção dos conjuntos A e B.

Diferença

Código:

```

conjuntoA = [1,2,3,4,5]
conjuntoB = [4,5,6,7,8]
conjuntoDiferencaAB = []
conjuntoDiferencaBA = []
for elementoDeA in conjuntoA:
if elementoDeA not in conjuntoB:
conjuntoDiferencaAB.append(elementoDeA)
for elementoDeB in conjuntoB:
if elementoDeB not in conjuntoA:
conjuntoDiferencaBA.append(elementoDeB)
print('A - B: ', conjuntoDiferencaAB)
print('B - A: ', conjuntoDiferencaBA)

```

Pseudo-Código:

Algoritmo 1: Diferença(A,B)

Entrada: A e B são conjuntos

Saída: O conjunto resultante da diferença entre o conjunto AB e BA

```

1: CAB ← {}
2: CBA ← {}
3: para cada a ∈ A faça:
4: | se a ∉ B então:
5: || Insere(CAB, a)
6: fim
7: para cada b ∈ B faça:
8: | se b ∉ A então:
9: || Insere(CBA,b)
10:fim
11: retorne "Conjunto AB:" CAB
12: retorne "Conjunto BA: " CBA

```

Explicação Código:

O algoritmo 1 realiza a Diferença dos conjuntos A e B, B e A.

Complemento

Codigo:

```

conjuntouniverso = [1,2,3,4,5,6,7,8,9]
conjuntoB = [1,2,3,4]
conjuntocomplementodeB = []
for elementouniverso in conjuntouniverso:
if elementouniverso not in conjuntoB:
conjuntocomplementodeB.append(elementouniverso)
print(conjuntocomplementodeB)

```

Pseudo-Codigo:

Entrada: Conjunto Universo[], Conjunto B[],ConjuntoComplementoB[]

Saida: O complemento do ConjuntoComplementoB no ConjuntoB

```

1: CU ← {}
2: CB ← {}
3: CPB ← {}
2: para cada u ∈ CU faça:
3: | se u ∉ B faça:
4: || Insere(B, u)
5: fim
6: retorne CPB

```

Explicação Codigo:

O algoritmo 1 realiza a junção de um elemento de universo dentro de um conjunto B.

## Produto Cartesiano

Codigo:

```
conjuntoA = [1,2,3]
conjuntoB = [4,5]
produtoCartesiano = []
for elementoDeA in conjuntoA:
for elementoDeB in conjuntoB:
produtoCartesiano.append([elementoDeA, elementoDeB])
print(produtoCartesiano)
```

Pseudo-Codigo:

Algoritmo 1: Produto Cartesiano(A,B)

Entrada: A e B são conjuntos

Saída: O conjunto resultante do produto cartesiano entre o conjunto A e B

```
1: PC ← {}
2: para cada a ∈ A faça:
3: | para cada b ∈ B faça:
4: || Insere(PC, a, b)
5: fim
6: retorne PC
```

Explicação Código:

O algoritmo 1 realiza uma Junção dos conjuntos A e B.

## União Disjunta

Codigo:

```
conjuntoA = [1,2,3]
conjuntoB = [3,4,5]
uniaoDisjunta = []
for elementoDeA in conjuntoA:
uniaoDisjunta.append([elementoDeA, 'conjuntoA'])
for elementoDeB in conjuntoB:
uniaoDisjunta.append([elementoDeB, 'conjuntoB'])
print(uniaoDisjunta)
```

Pseudo-Codigo:

Algoritmo 1: União Disjunta(A,B)

Entrada: A e B são conjuntos

Saída: O Conjunto resultante da união dos conjuntos neles mesmos

```
1: CUD ← {}
2: para cada a ∈ A faça:
```

```
3: | Insere(CUD,a, "conjunto a")
4: para cada  $b \in B$  faça:
5: | Insere(CUD, b, "conjunto b")
6: fim
7: retorne CUD
```

#### Explicação Código:

O algoritmo 1 realiza uma Junção dos conjuntos A e B do qual os elementos de um não vão para o outro.