

**Algoritmo 1:** União(A,B)**Entrada :** A e B são conjuntos**Saída :** O conjunto resultante da união entre o conjunto A e B

```
1.  $U \leftarrow \{\}$ 
2. maior  $\leftarrow$  o maior conjunto entre A e B
3. menor  $\leftarrow$  o menor conjunto entre A e B
4. para cada  $a \in$  maior faça
5. | insere(U,a)
6. fim
7. para cada  $b \leftarrow$  menor faça
8. | se  $b \notin U$ 
9. || insere(U,b)
10. fim
11. retorna U
```

**Código:**

```
1. A = [1,2,3,5,6]
2. B = [1,2,3,4,5,6,7,8,9,0]
3. uniao = []
4. maior = max(A,B)
5. menor = min(A,B)
6. for x in maior:
7. | uniao.append(x)
8. for x in menor:
9. | if x not in uniao:
10. || uniao.append(x)
11. print(uniao)
```

**Algoritmo 2:** Intersecção(A,B)**Entrada:** A e B são conjuntos**Saída:** O conjunto resultante da intersecção entre o conjunto A e B

```
1. I  $\leftarrow \{\}$ 
2. maior  $\leftarrow$  o maior conjunto entre A e B
3. menor  $\leftarrow$  o menor conjunto entre A e B
4. para cada  $a \in$  maior faça
5. | se  $a \in$  menor
6. || se  $a \notin I$ 
7. ||| insere(I,a)
```

8. fim
9. retorna l

**Código:**

```
1.A = [0,8,1,2,3,5,6]
2.B = [1,2,3,4,5,5,5,5,6]
3.maior = max(A,B)
4.menor = min(A,B)
5.terseccao = []
6.for x in maior:
7. | if x in menor:
8. || if x not in terseccao:
9. ||| terseccao.append(x)
10.print(terseccao)
```

**Algoritmo 3:** Diferença(A,B)

**Entrada:** A e B são conjuntos

**Saída:** O conjunto resultante da diferença entre o conjunto A e B

1.  $D \leftarrow \{\}$
2. maior  $\leftarrow$  o maior conjunto entre A e B
3. menor  $\leftarrow$  o menor conjunto entre A e B
4. para cada  $a \in$  maior faça
5. | se  $a \in$  maior e  $a \notin$  menor
6. || se  $a \notin D$
7. ||| insere(D,a)
8. fim
9. retorna D

**Código:**

```
1. A = [0,1,2,3,4,5,6]
2. B = [2,4,6,8]
3. menor = max(A,B)
4. maior = min(A,B)
5. diferenca = []
6. for x in maior:
7. | if x in maior and x not in menor:
8. || if x not in diferenca:
9. ||| diferenca.append(x)
```

```
10. print(diferenca)
```

**Algoritmo 4:** Complemento**Entrada:** A e B são conjuntos**Saída:** O conjunto resultante do complemento entre o conjunto A e B

```
1. C ← {}
2. maior ← o maior conjunto entre A e B
3. menor ← o menor conjunto entre A e B
4. para cada a ∈ menor faça
5. | se a ∉ maior
6. || se a ∉ C
7. ||| insere(C,a)
8. fim
9. retorna C
```

**Código:**

```
1. A = [1,6,7]
2. B = [1,2,3,4,5,5,5,6]
3. maior = max(A,B)
4. menor = min(A,B)
5. complemento = []
6. for x in menor:
7. | if x not in maior:
8. || if x not in complemento:
9. ||| complemento.append(x)
10. print(complemento)
```

**Algoritmo 5:** ConjuntoDasPartes**Algoritmo 6:** ProdutoCartesiano(A,B)**Entrada:** A e B são conjuntos**Saída:** O conjunto resultante do arranjo entre o conjunto A e B

```
1. C ← {}
2. para cada a ∈ A faça
3. | para cada b ∈ B faça
4. || insere(C,a,b)
5. | fim
6. fim
7. retorne C
```

**Código:**

```
1.A = ["a","b","c"]
2.B = [0,1,2]
3.produtocartesiano = []
4.for x in A:
5. | for y in B:
6. || produtocartesiano.append((x,y))
7.print(produtocartesiano)
```

**Algoritmo 7:** União disjunta(A,B)

**Entrada:** A e B são conjuntos

**Saída:** O conjunto resultante da disjunção entre o conjunto A + B

```
1.Ud ← {}
2.para cada a ∈ A faça
3. | insere(Ud,a,"A")
4.para cada a ∈ B faça
5. | insere(Ud,a,"B")
6. retorna Ud
```

**Código:**

```
1. A = [5,2,3]
2. B = [3,4,5]
3. udisjunta = []
4. for x in A:
5. | udisjunta.append([x, 'A'])
6. for x in B:
7. | udisjunta.append([x, 'B'])
8. print(udisjunta)
```