

```

1- buscar-binario (V[], inicio, fim, i)
    e = (inicio + fim) / 2
    if (V[i] == i)
        return i
    if (inicio == fim)
        elif (V[i] < e)
            buscar-binario (V, i+1, fim, e)
        else
            buscar-binario (V, inicio, i+1, i)

```

```

2- def merge-sort (A):

```

```

    if len (A) <= 1

```

```

        return A, 0

```

```

    meio = len (A) // 2

```

```

    esq, inv-esq = merge-sort (A[:meio])

```

```

    dir, inv-dir = merge-sort (A[meio:])

```

```

    ordenado, inv-merge = no contar-inversoes (esq, dir)

```

```

    return ordenado, inv-esq + inv-dir + inv-merge

```

```

def contar-inversoes (esq, dir):

```

```

    ordenado = []

```

```

    inversoes = 0

```

```

    i = j = 0

```

```

    while i < len(esq) and j < len(dir):

```

```

        if esq[i] <= dir[j]:

```

```

            ordenado.append (esq[i])

```

```

            i += 1

```

```

        else:
            ordenado.append (dir[j])

```

```

            j += 1

```

```

            inversoes += len(esq) - i

```

```

    ordenado.extend (esq[i:])

```

```

    ordenado.extend (dir[j:])

```

```

    return ordenado, inversoes

```