array 🡨 array original que es dado

max\_length 🡨 cantidad de dígitos del mayor numero del array

auxiliar 🡨 array auxiliar para trabajar los números como sublistas

buckets 🡨 array temporal de números ordenados con counting sort

auxiliar\_buckets 🡨 array que copia a buckets

otro\_mas 🡨 array acumulador de buckets y auxiliar\_buckets

lista\_resultante 🡨 array resultante

max\_num = EncontrarMaximo(array)

max\_length = LongitudDeCadena(max\_num)

auxiliar = InicializarArreglo(max\_length)

Para cada elemento en array:

elemento = ConvertirACadenaYAgregarCerosALaIzquierda(elemento, max\_length)

auxiliar[i] = elemento

auxiliar = ConvertirAArregloDeArreglos(auxiliar)

otro\_mas = [ ]

Para cada posición de dígito desde la Izquierda hasta la derecha:

buckets = InicializarArregloDeArreglos(10)

buckets\_buckets= InicializarArregloDeArreglos(10)

Para cada subarreglo en auxiliar:

Si la posición del digito es diferente de cero:

buckets = InicializarArregloDeArreglos(10)

Para cada elemento en subarray:

Si subarreglo no está vacío:

dígito = ObtenerDígitoEnPosición(subarreglo, posición)

Agregar subarreglo a buckets[dígito]

Si es el último elemento en el subarreglo:

Si posición no es cero:

otro\_mas = CombinarArreglos(auxiliar\_buckets, buckets)

auxiliar\_buckets = otro\_mas

Sino:

CopiarValoresDeArreglo( de: buckets, a: auxiliar\_buckets)

Reiniciar auxiliar

CopiarValoresDeArreglo(de: auxiliar\_buckets, a: auxiliar)

lista\_resultante = [ ]

Para cada subarreglo en auxiliar:

Para cada elemento en subarreglo:

Si elemento no contiene solo ceros a la izquierda:

Agregar elemento a lista\_resultante después de eliminar ceros a la izquierda

Imprimir(lista\_resultante)