

num contar (num n)

IF ($n = 0$) THEN
RETURN 0

arregloCadenas listaCadenas
num contCadenas = 10^{**n}

FOR $i=0$ TO contCadenas - 1
listaCadenas[i] = carácter (i) // Transformar a tipo carácter

num cadenasValidas = 0

FOR $j=0$ TO contCadenas - 1

num digitos = 0

num cadena = j

IF (cadena = 0) THEN

digitos = 1

ELSE

WHILE (cadena ≥ 1)

cadena = cadena / 10

digitos = digitos + 1

num cantCeros = 0

FOR $k=0$ TO digitos - 1

IF (listaCadenas[i][j] = 0) THEN

cantCeros = cantCeros + 1

// Carácter 0

IF (((cantCeros + (n - digitos)) % 2 = 1) THEN

cadenasValidas = cadenasValidas + 1

RETURN cadenasValidas

$$\therefore T(n) = c + 3c + n + c + \sum_{i=1}^n (3c + \sum_{j=1}^i (4c) + c + \sum_{k=1}^j (3c) + 6c) + c$$

$$T(n) = 6c + 9cn + n \cdot \sum_{i=1}^n (3c) + n \sum_{i=1}^n (4c)$$

$$T(n) = 6c + 9cn + 7cn^2$$

$$\Rightarrow O(n^2)$$

arregloNum unionSR (arregloNum A, arregloNum B)

num largoA = largo (A)

num largoB = largo (B)

num indiceC = 1

arregloNum C

QuickSort (A, 1, largoA)

QuickSort (B, 1, largoB)

FOR i=1 TO largoA

IF (busquedaBinaria (B, 1, largoB, A[i]) = F) THEN

C[indiceC] = A[i]

indiceC = indiceC + 1

FOR j=1 TO largoB

IF (busquedaBinaria (C, 1, indiceC, B[j]) = F) THEN

C[indiceC] = B[j]

indiceC = indiceC + 1

RETURN C

$$T(n) = 4c + n \cdot \log_2 n + n \cdot \log_2 n + \sum_1^n (\log_2 n) + \sum_1^n (\log_2 n) + c$$

$$T(n) = 5c + 2n \cdot \log_2 n + n \cdot \log_2 n + n \cdot \log_2 n$$

$$T(n) = 5c + 4n \cdot \log_2 n$$

$$\Rightarrow O(n \cdot \log_2 n)$$