```
MERGE(A, p, q, r)
1.  nL = q - p + 1          // length of A[p:q]
2.  nR = r - q
3.  let L[0:nL-1] and R[0:nR-1] be new arrays
4.  for i = 0 to nL - 1     // copy A[p:q] into L[0:nL-1]
5.      L[i] = A[p+i]
6.  for j = 0 to nR - 1     // copy A[q+1:r] into R[0:nR-1]
7.      R[j] = A[q+j+1]
8.  i = 0
9.  j = 0
10. k = p
11. // As long as each of the arrays L and R contains an
    // unmerged element, copy the smallest unmerged element back into
12. while i < nL and j < nR
13.     if L[i] ≤ R[j]
14.         A[k] = L[i]
15.         i = i + 1
16.     else A[k] = R[j]
17.         j = j + 1
18.     k = k + 1
19. // Having gone through one of L and R entirely, copy
20. // the remainder of the other to the end of A[p:r]
21.     while i < nL
22.         A[k] = L[i]
23.         i = i + 1
```

```
24.                 k = k + 1
25          While   j < nR
26                  A[k] = R[j]
27                  j = j + 1
28.                 k = k + 1
```

Datos:

A = [2, 4, 5, 1, 3, 6]

p = 0

q = 2

r = 5

Inicia

n_L = 2 - 0 + 1 = 3

n_R = 5 - 2 = 3

L = [2, 4, 5]

R = [1, 3, 6]

Indice Inicial

i = 0

j = 0

k = 0

// Ciclo

interación	L[i]	R[j]	A[k]	Resultado parcial en A	i	j	k
1.	2	1	R[j]=1	[1,4,5,1,3,6]	0	1	1
2	2	3	L[i]=2	[1,2,5,1,3,6]	1	1	2
3	4	3	R[j]=3	[1,23,1,3,6]	1	2	3
4	4	6	L[i]=4	[1,2,3,43,6]	2	2	4
5	5	6	L[i]=5	[1,2,3,4,5,6]	3	2	5

Ciclo 19 - 22 . L .(i=3, n_L=3) → No entra

Ciclo .23 - 27 R .(j=2, n_R=3)

- interacción :
  - A[5] = R[2] = 6 → ya está ahí
  - j=3 , k=6

Resultado Final:

A = [1, 2, 3, 4, 5, 6]