Contents

1)

Nivel	llamada	rgt > lft	mid	a[lft..rgt]
0	ms_rec(a, 1, 7)	True	4	[7, 1, 10, 3, 4, 9, 5]
1	ms_rec(a, 1, 4)	True	2	[7, 1, 10, 3]
2	ms_rec(a, 1, 2)	True	1	[7, 1]
3	ms_rec(a, 1, 1)	False		[7]
3	ms_rec(a, 1+1, 2)	False		[1]
2	merge(a, 1, 1, 2)		1	[1, 7]
2	ms_rec(a, 2+1, 4)	True	3	[10, 3]
3	ms_rec(a, 3, 3)	False	1	[10]
3	ms_rec(a, 3+1, 4)	False	1	[3]
2	merge(a, 3, 3, 4)		3	[3, 10]
0	ms_rec(a, 4+1, 7)	True	6	[4, 9, 5]
1	ms_rec(a, 5, 6)	True	5	[4, 9]
2	ms_rec(a, 5, 5)	False		[4]
2	ms_rec(a, 5+1, 6)	False		[9]
1	merge(a, 5, 5, 6)		5	[4, 9]
1	ms_rec(a, 6+1, 7)	False		[5]
1	merge(a, 1, 2, 4)		2	[1, 3, 7, 10]
0	merge(a, 5, 6, 7)		6	[4, 5, 9]
0	merge(a, 1, 4, 7)		4	[1, 3, 4, 5, 7, 9, 10]

2)

a)

Prueba:

$$\{i := 0, a := [3,7,1,6,1,5,3,4]\}$$
$$merge(a[1, 2^0], a[2^0 + 1, 2 * 2^0])$$
$$merge(a[2 * 2^0 + 1, 3 * 2^0], a[3 * 2^0 + 1, 4 * 2^0])$$
$$\equiv \{\text{Aritmetica}\}$$
$$merge(a[1, 1], a[1 + 1, 2 * 1])$$
$$merge(a[2 * 1 + 1, 3 * 1], a[3 * 1 + 1, 4 * 1])$$
$$\equiv \{\text{Aritmetica}\}$$
$$merge(a[1, 1], a[2, 2])$$
$$merge(a[3, 3], a[4, 4])$$
$$\{a := [\mathbf{3}, \mathbf{7}, \mathbf{1}, \mathbf{6}, 1, 5, 3, 4]\}$$

$$\{i := 1, a := [3,7,1,6,1,5,3,4]\}$$
$$merge(a[1, 2^1], a[2^1 + 1, 2 * 2^1])$$
$$merge(a[2 * 2^1 + 1, 3 * 2^1], a[3 * 2^1 + 1, 4 * 2^1])$$
$$\equiv \{\text{Aritmetica}\}$$
$$merge(a[1, 2], a[2 + 1, 2 * 2])$$
$$merge(a[2 * 2 + 1, 3 * 2], a[3 * 2 + 1, 4 * 2])$$
$$\equiv \{\text{Aritmetica}\}$$
$$merge(a[1, 2], a[3, 4])$$
$$merge(a[5, 6], a[7, 8])$$
$$\{a := [\mathbf{1}, \mathbf{3}, \mathbf{6}, \mathbf{7}, \mathbf{1}, \mathbf{3}, \mathbf{4}, \mathbf{5}]\}$$
$$\vdots$$

Formula general:

$merge(a, \underbrace{j * 2^i + 1}_{lft}, \underbrace{(j+1) * 2^i}_{mid}, \underbrace{(j+2) * 2^i}_{rgt}),\ \text{ con } j = 0, 2, \ldots, \frac{n}{2^i}$

Procedimiento:

```
proc intercalarCada(in/out a: array[1..$2^n$] of int, in i: nat)
  var lft, rgt, mid, j: nat
  j := 0
  while $j \leq 2^n$ do
    lft := $j * 2^i + 1$
    mid := $(j + 1) * 2^i$
    rgt := $(j + 2) * 2^i$
    merge(a, lft, mid, rgt)
    j:= j + 2
  od
end proc
```

b)

```
proc iterMerge(in/out a: array[1..$2^n$] of int)
  for i := 0 to $n - 1$ do
    intercalarCada(a, i)
  od
end proc
```

3)

a)

Nivel	Llamada	rgt > lft	ppiv	a[lft..rgt]
0	qs_rec(a, 1, 7)	True		[7, 1, 10, 3, 4, 9, 5]
0	partition(a, 1, 7, ppiv)		5	[1, 3, 4, 5] [7] [10, 9]
0	qs_rec(a, 1, 5-1)	True		[1, 3, 4, 5]
1	partition(a, 1, 4, ppiv)		1	[] [1] [3, 4, 5]
1	qs_rec(a, 1, 1-1)	False		[]
1	qs_rec(a, 1+1, 4)	True		[3, 4, 5]
2	partition(a, 2, 4, ppiv)		2	[] [3] [4, 5]
2	qs_rec(a, 2, 2-1)	False		[]
2	qs_rec(a, 2+1, 4)	True		[4, 5]
3	partition(a, 3, 4, ppiv)		3	[] [4] [5]
3	qs_rec(a, 3, 3-1)	False		[]
3	qs_rec(a, 3+1, 4)	False		[5]
0	qs_rec(a, 5+1, 7)	True		[10, 9]
1	partition(a, 6, 7, ppiv)		6	[] [9] [10]
1	qs_rec(a, 6, 6-1)	False		[]
1	qs_rec(a, 6+1, 7)	False		[10]
Final				[1, 3, 4, 5, 7, 9, 10]