



Ordenação de dados

Glauco Vinicius Scheffel, Msc



"Poderia viver encerrado numa casca de noz e julgar-me rei do infinito..." Shakespeare - Hamlet, ato 2, cena 2



Dividir e Conquistar

“O problema é subdividido em um ou mais subproblemas, cada um similar ao problema original. Cada um dos subproblemas é resolvido independentemente. Finalmente, as soluções dos subproblemas são combinadas a fim de se obter a solução do problema original.”

Bruno R. Preiss



Dividir e conquistar

1. Problema pequeno, resolva o problema
2. Dividir o problema em dois ou mais problemas menores e similares
3. Recursão resolva o problema
4. Conquistar, reuna as soluções parciais unindo as seqüências



Merge Sort

Visão Geral



Visão geral Merge Sort





Visão geral Merge Sort

Parte I

Parte 2

dividir



Visão geral Merge Sort

Parte 2

Parte I

conquistar



Merge Sort
Conquista



Merge (fase conquista)

19

2

43

30

15

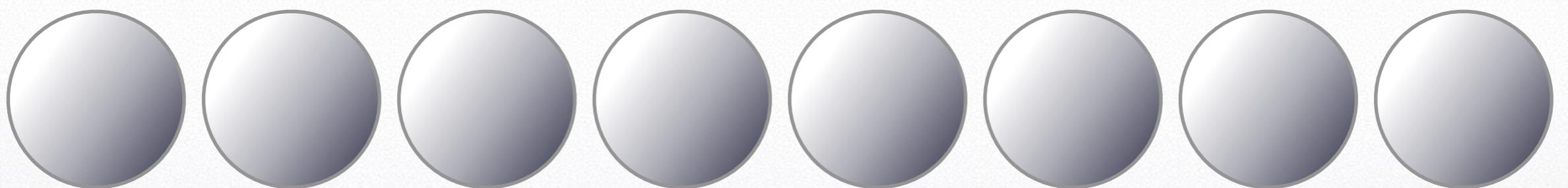
12

51

32



Merge (fase divisão)



19

2

43

30

15

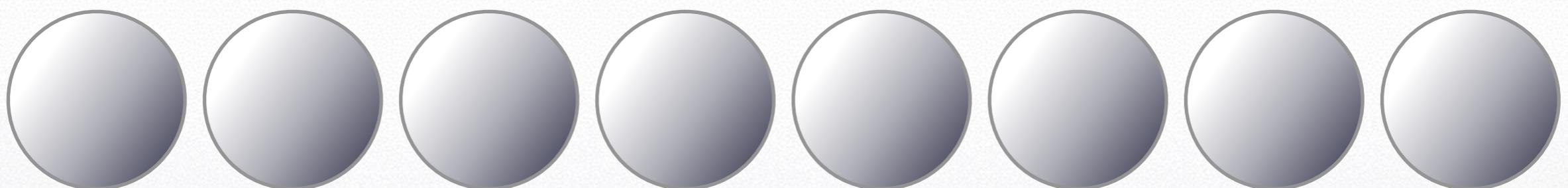
12

51

32



Merge (fase divisão - ordenar)



2

19

30

43

15

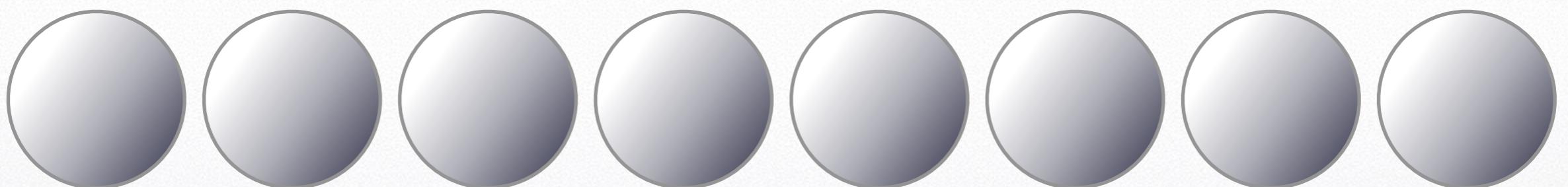
12

51

32



Merge (fase divisão - ordenar)



2

19

30

43

12

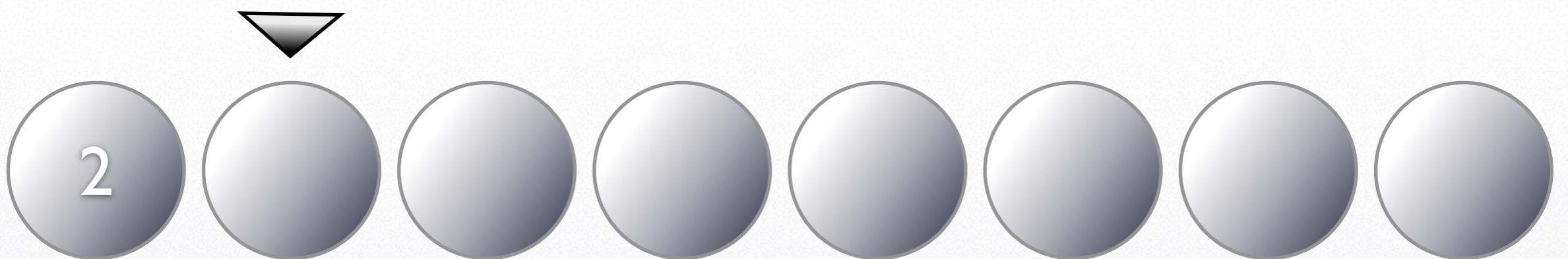
15

32

51

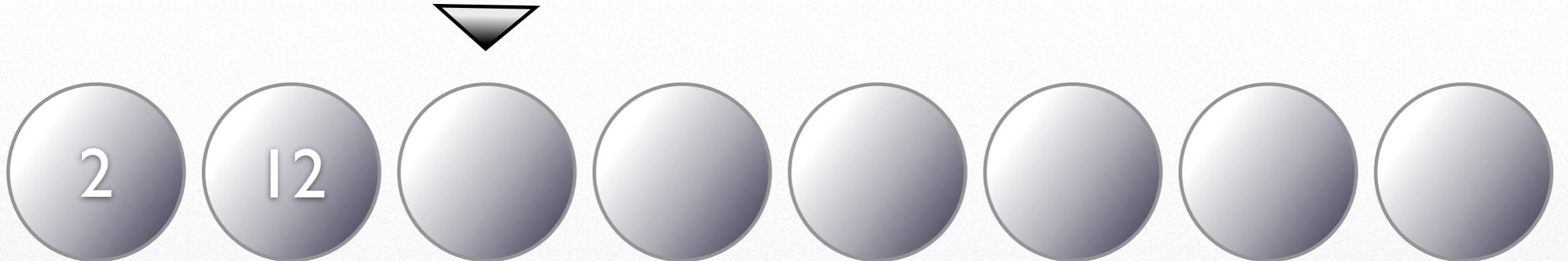


Merge (fase conquista)





Merge (fase conquista)





Merge (fase conquista)

2

12

15

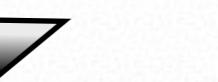


2

19

30

43



12

15

32

51





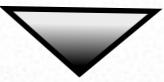
Merge (fase conquista)

2

12

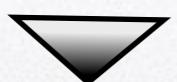
15

19



2

19



2

19

30

43

12

15

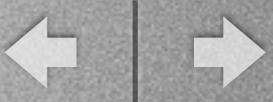
32

51



Merge (fase conquista)





Merge (fase conquista)

2

12

15

19

30

32



2

19

30

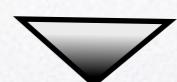
43

12

15

32

51





Merge (fase conquista)

2

12

15

19

30

32

43



2

19

30

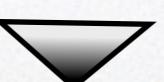
43

12

15

32

51





Merge (fase conquista)

2

12

15

19

30

32

43

51



2

19

30

43

12

15

32

51



Merge Sort Algoritmo



Merge Sort: Algoritmo

Merge-Sort (array, esquerda, direita)

se esquerda \geq direita então

retornar

senão

area = esquerda+direita

meio = arredondar(area / 2)

Merge-Sort(array, esquerda, meio)

Merge-Sort(array, meio+1, direita)

Merge(array, esquerda, meio, direita)





Merge: Algoritmo



função **Merge**(A, esquerda, meio, direita)

$n1 = \text{meio} - \text{esquerda} + 1$

$n2 = \text{direita} - \text{meio}$

criar array temporario L[$n1$], R[$n2$]

for $i = 0$ to $n1-1$ faça

$L[i] = A[\text{esquerda} + i]$

for $j = 0$ to $n2-1$ faça

$R[j] = A[\text{meio}+j]$

zerar variáveis contadoras do loop



Merge: Algoritmo

enquanto $i < n1 \ \&\& \ j < n2$ faça

 se $L[i] < R[j]$ então

 array[k ++] = ESQUERDA[i ++]

 senão

 array[k ++] = DIREITA[j ++]

enquanto $i < n1$ faça

 direita[k ++] = ESQUERDA[i ++]

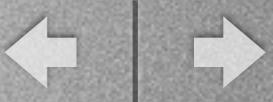
enquanto $j < n2$ faça

 array[k ++] = DIREITA[j ++]





Merge Sort
Divisão

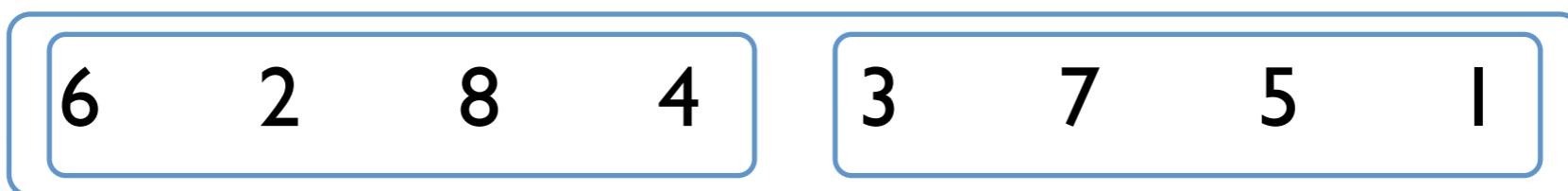


Merge (fase dividir)

6 2 8 4 3 7 5 |



Merge (fase dividir)



Merge-Sort(A, 0, 7),

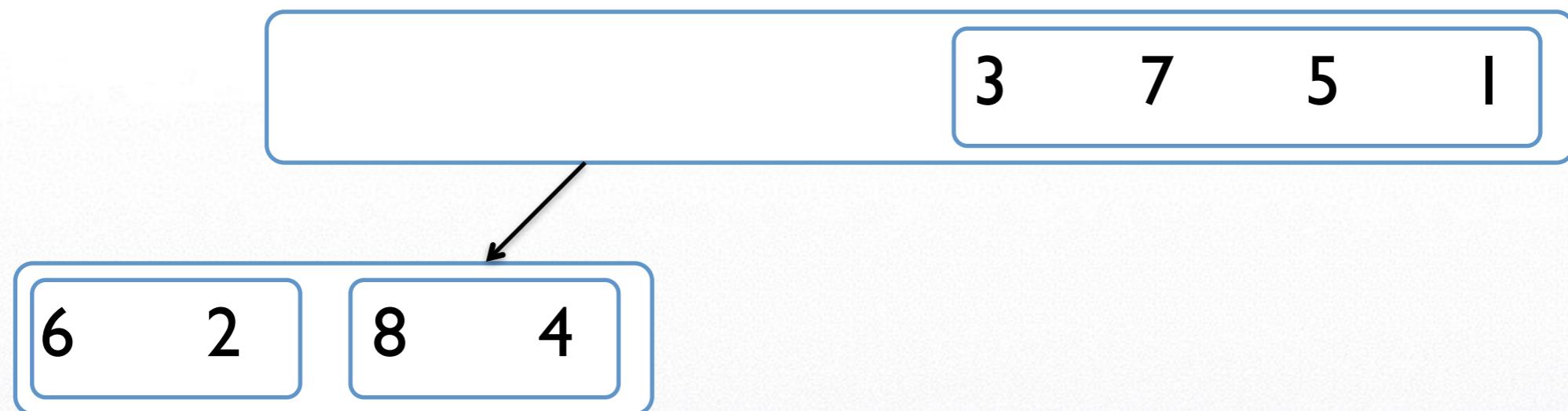


Merge (fase dividir)

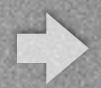




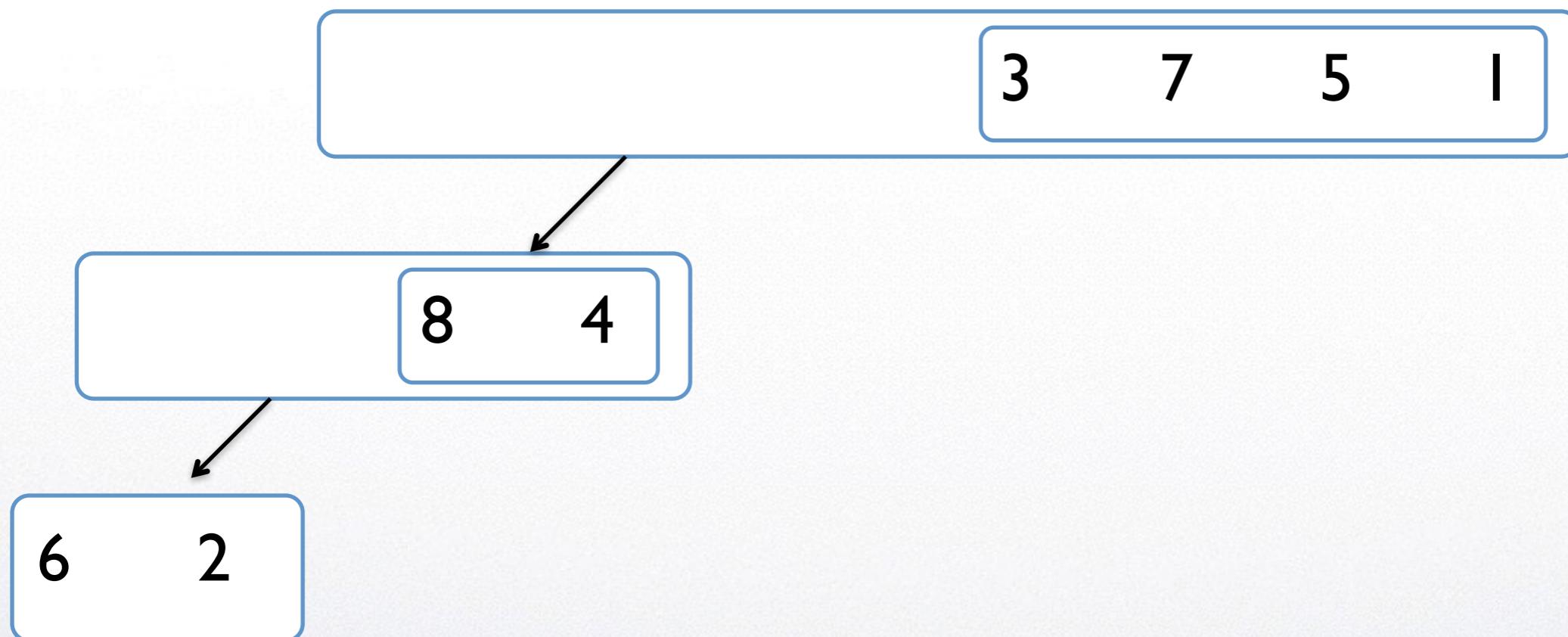
Merge (fase dividir)



Merge-Sort($A, 0, 7$), Merge-Sort($A, 0, 3$) dividir



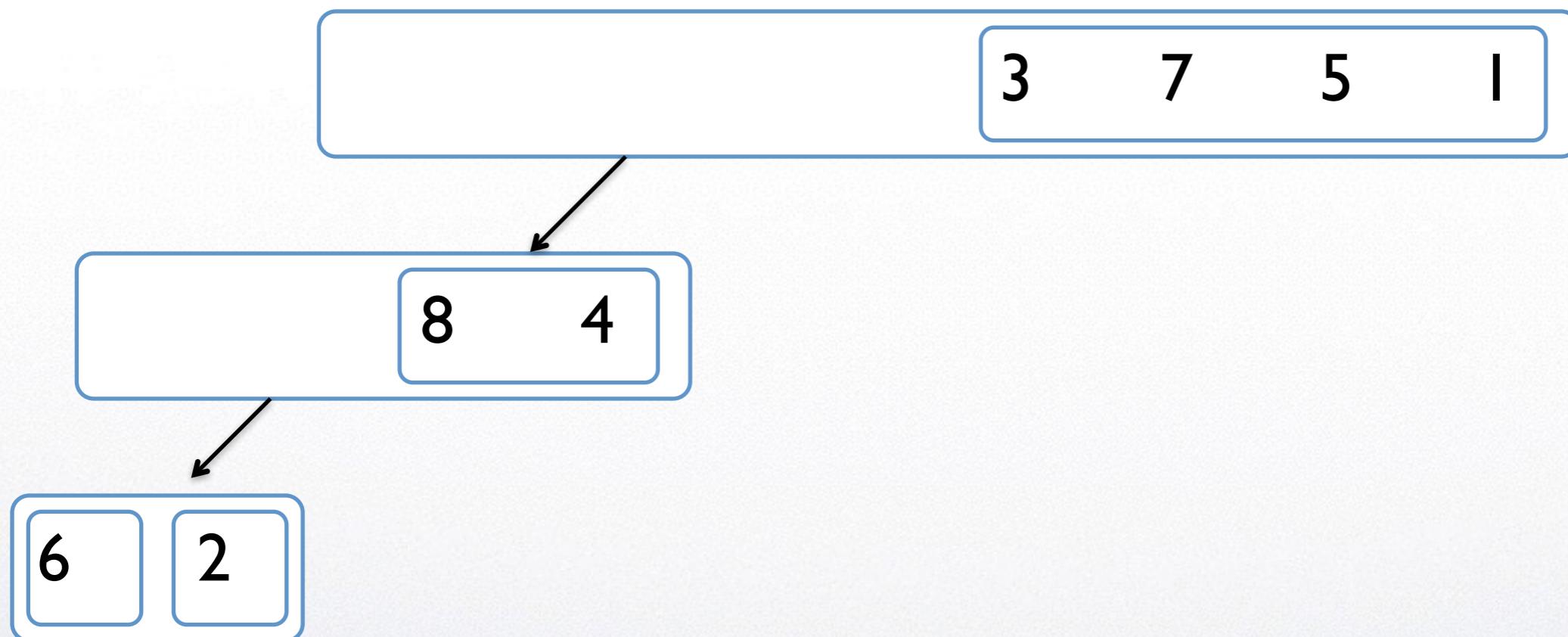
Merge (fase dividir)



Merge-Sort(A, 0, 7), Merge-Sort(A,0,1) dividir

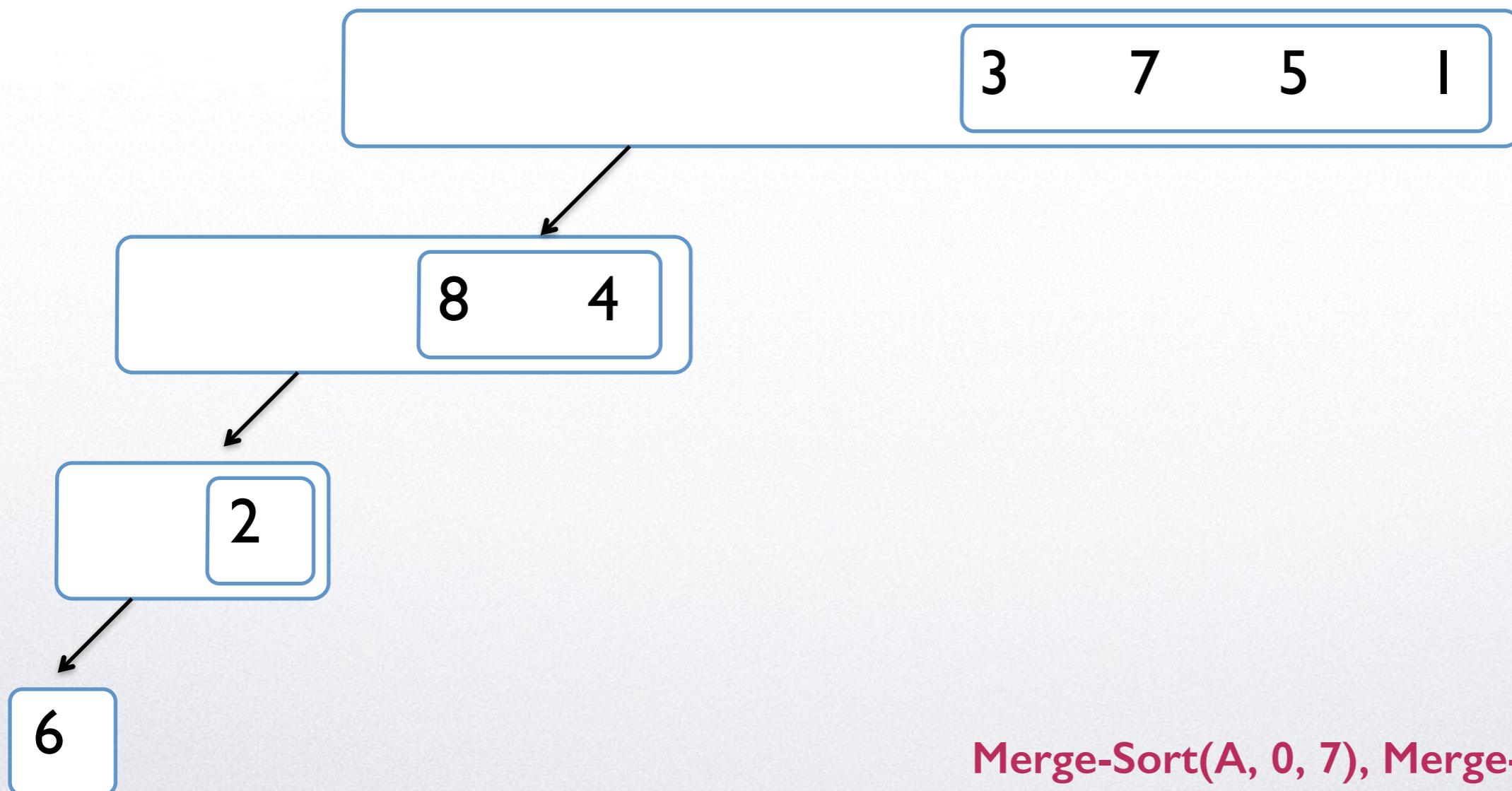


Merge (fase dividir)



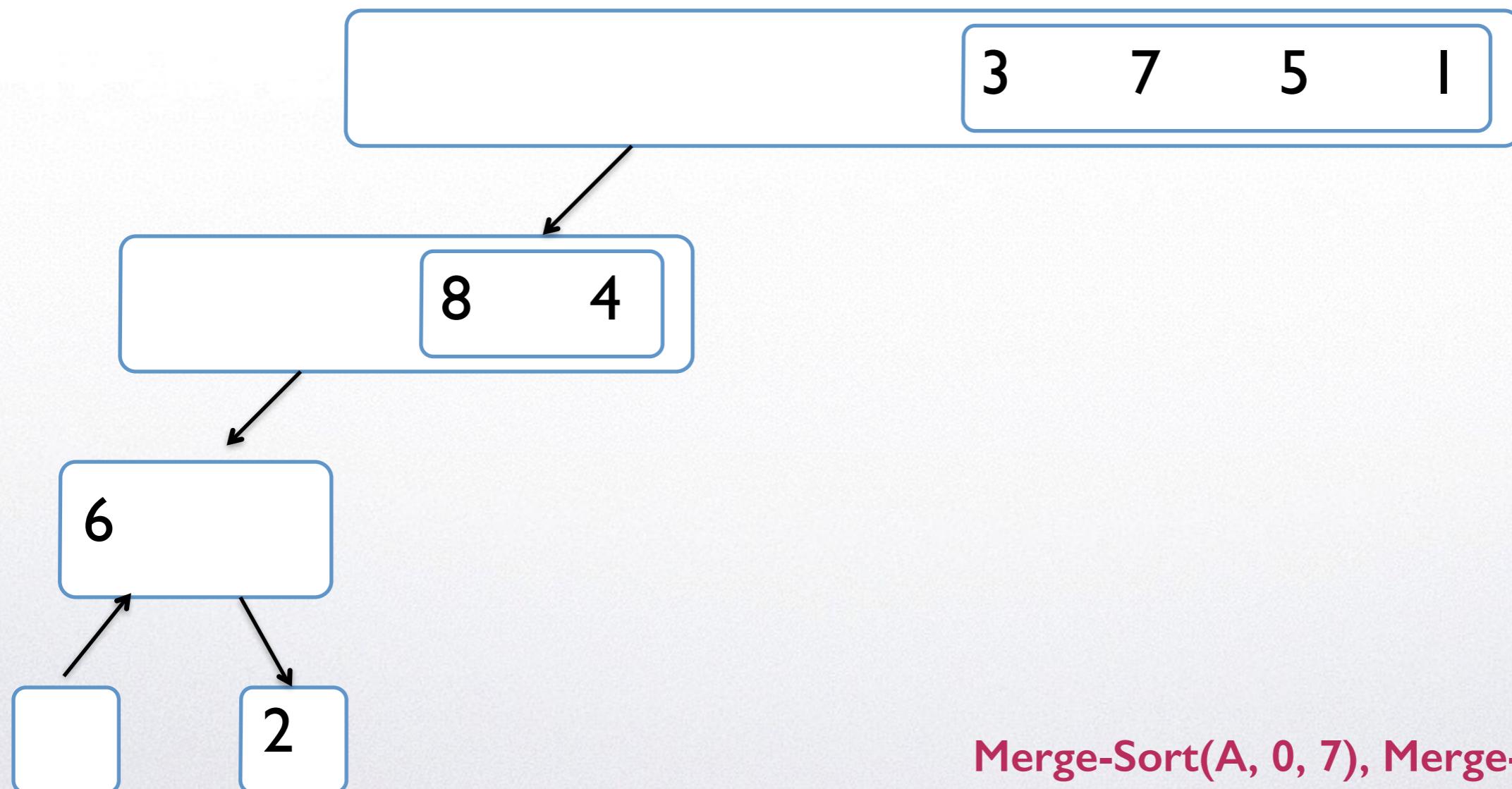


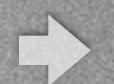
Merge (fase dividir)



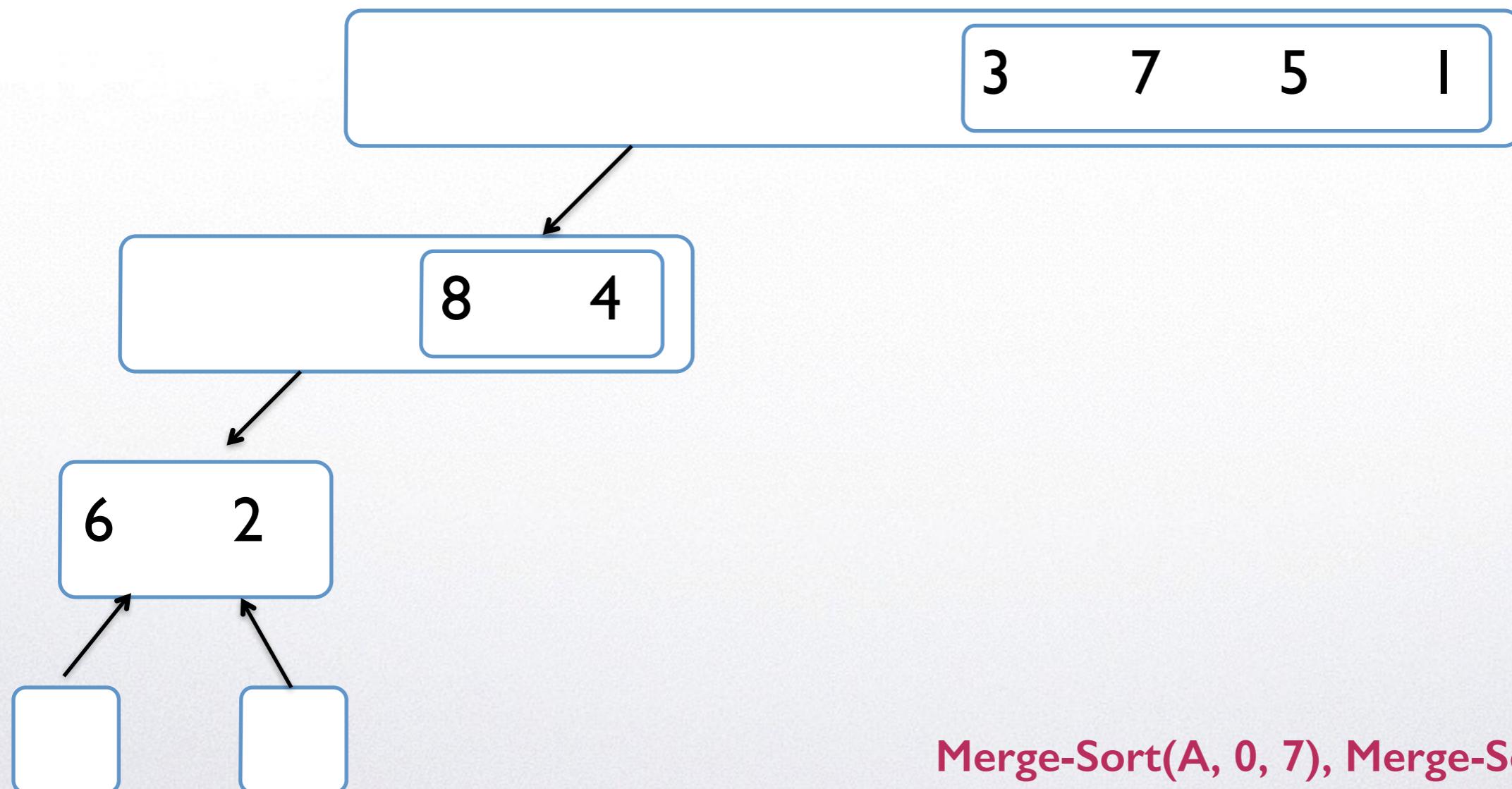


Merge (fase dividir)



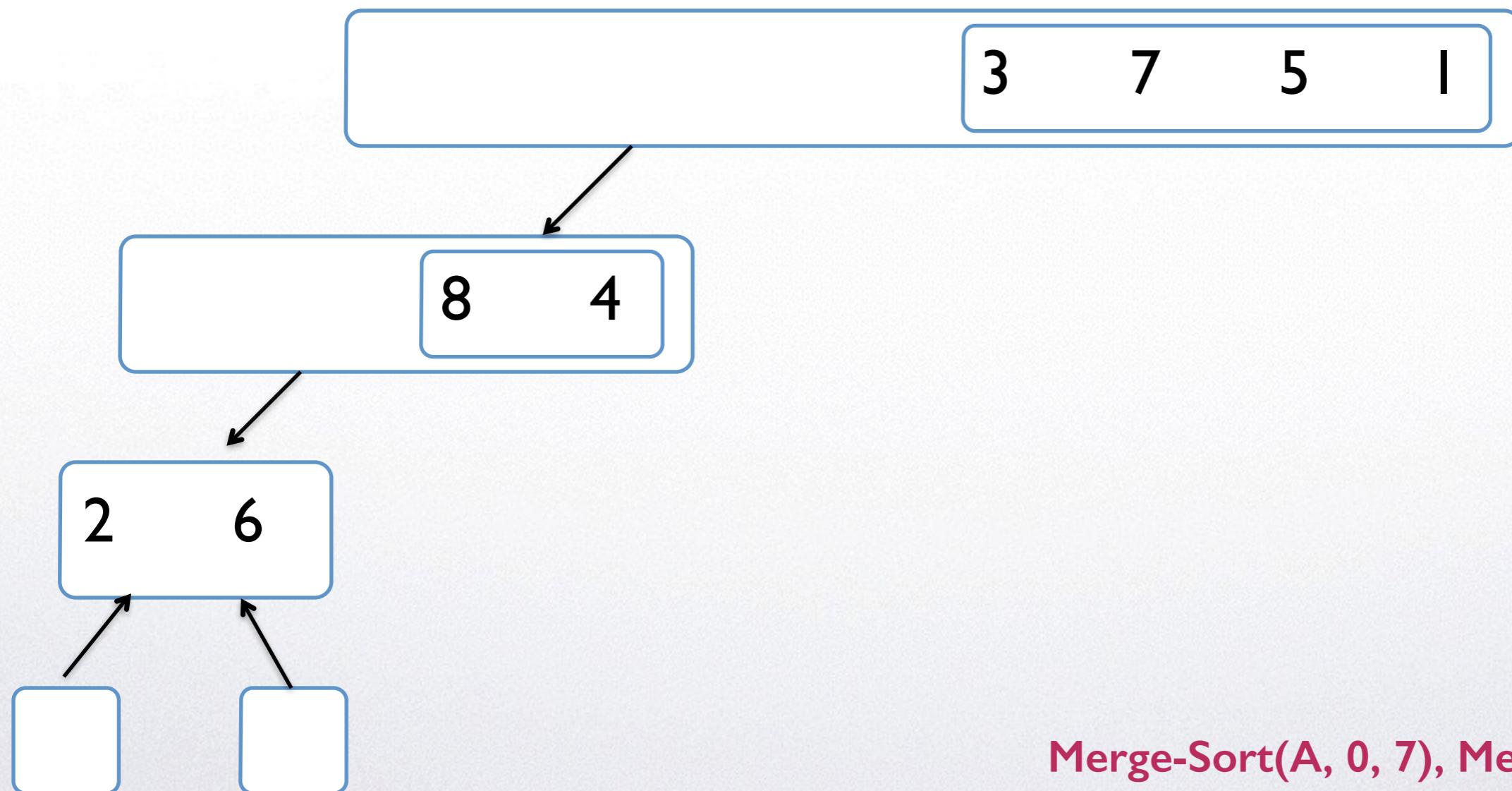


Merge (fase dividir)



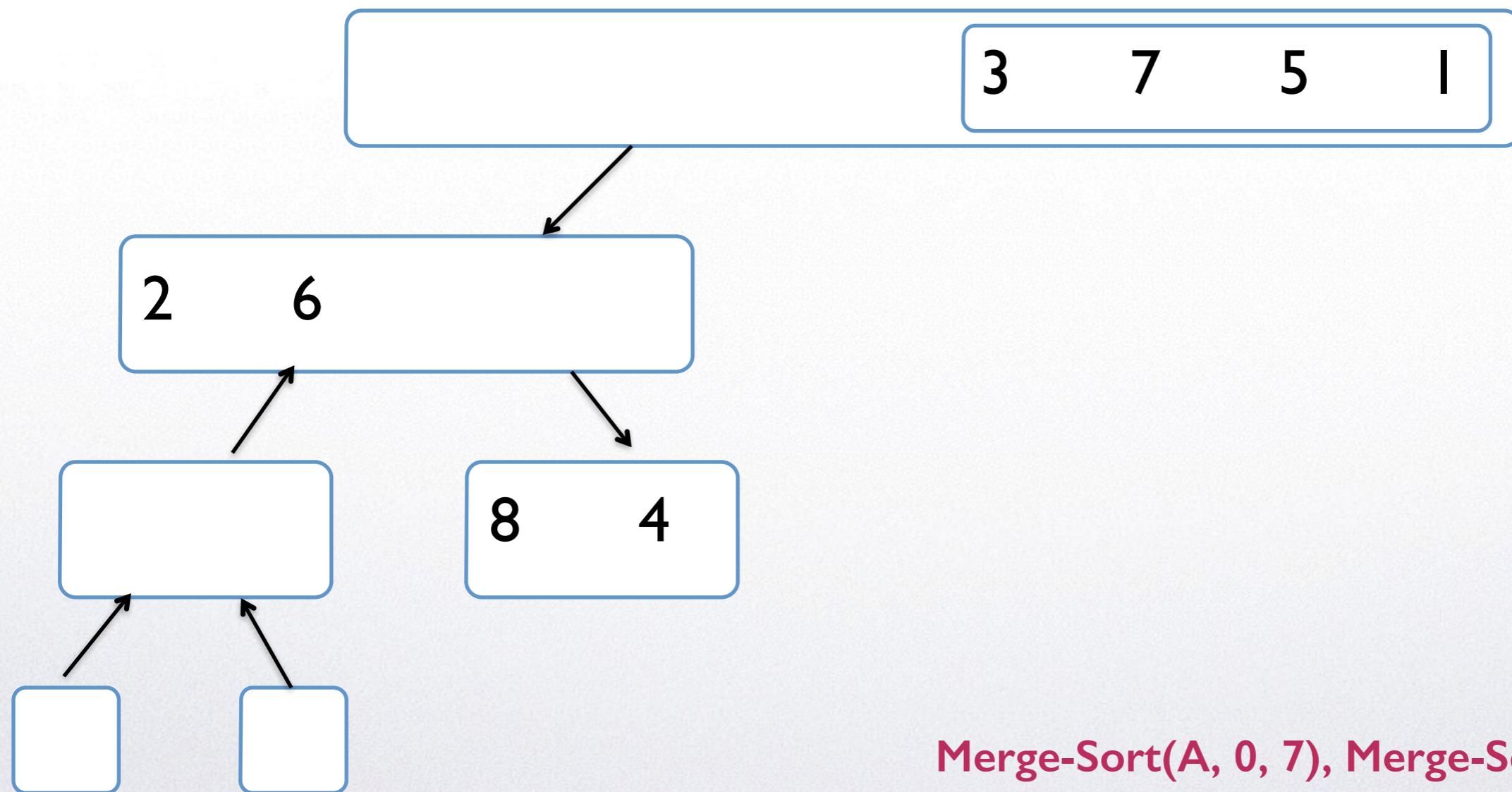


Merge (fase dividir)



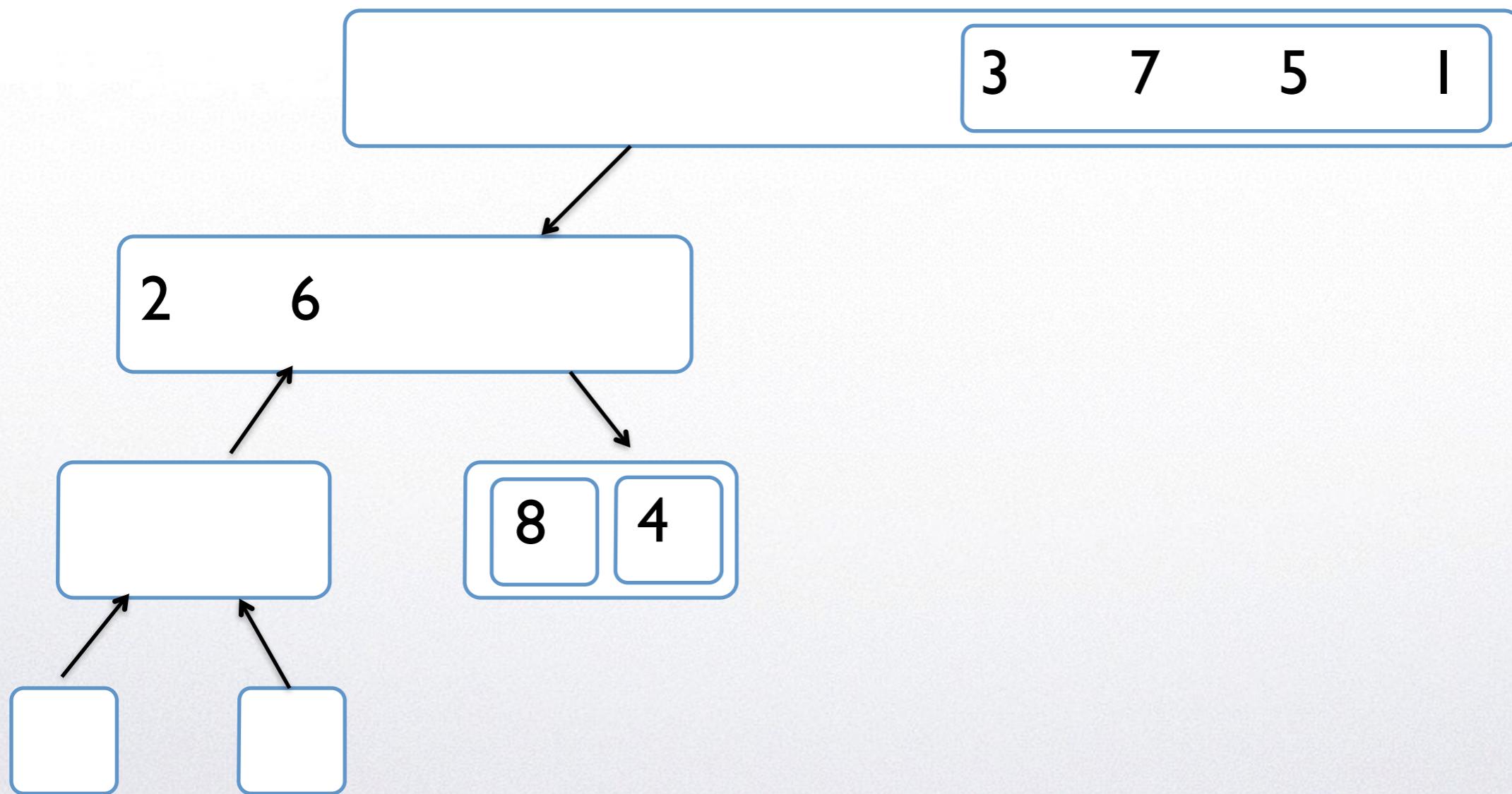


Merge (fase dividir)



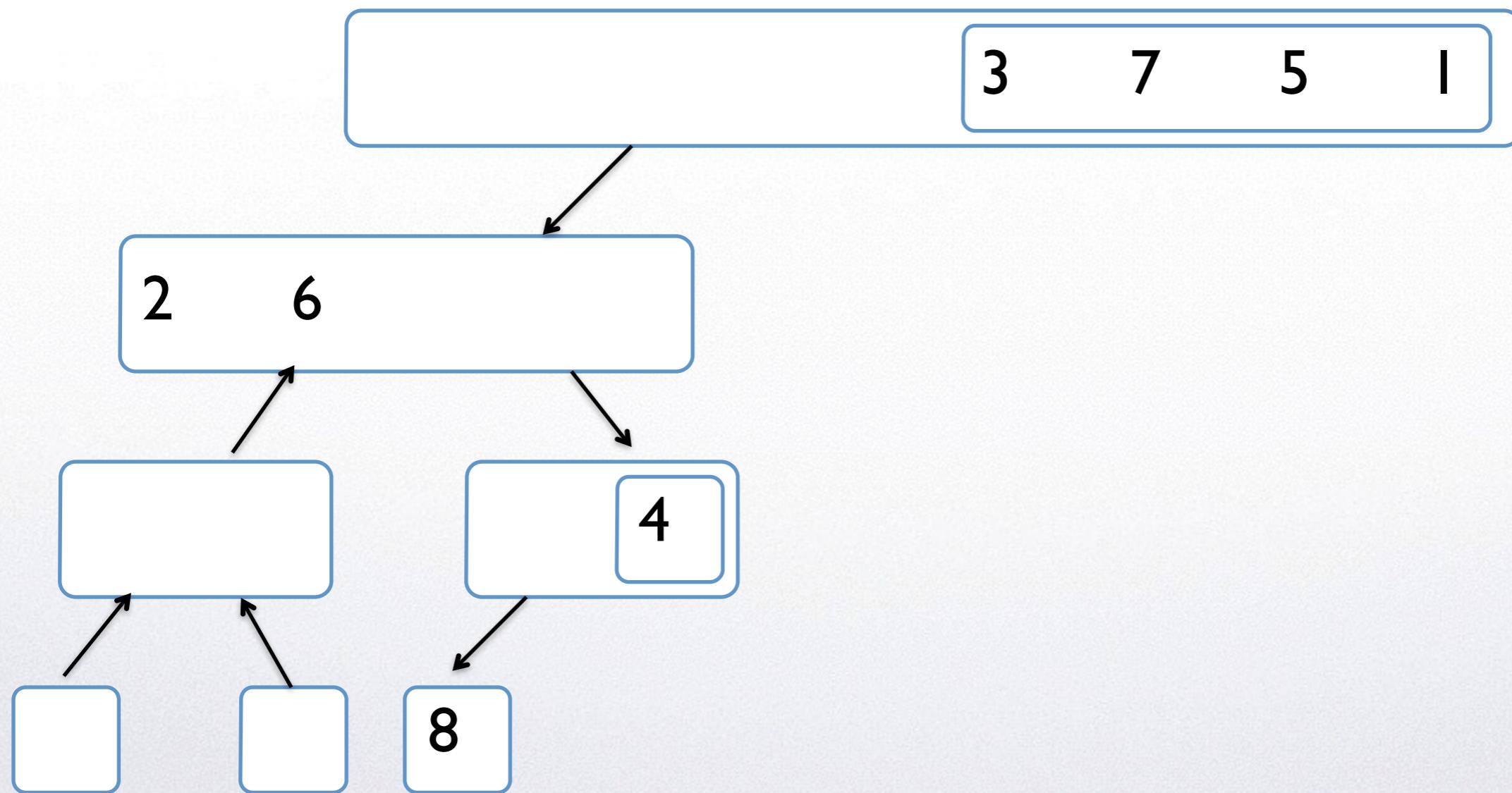


Merge (fase dividir)



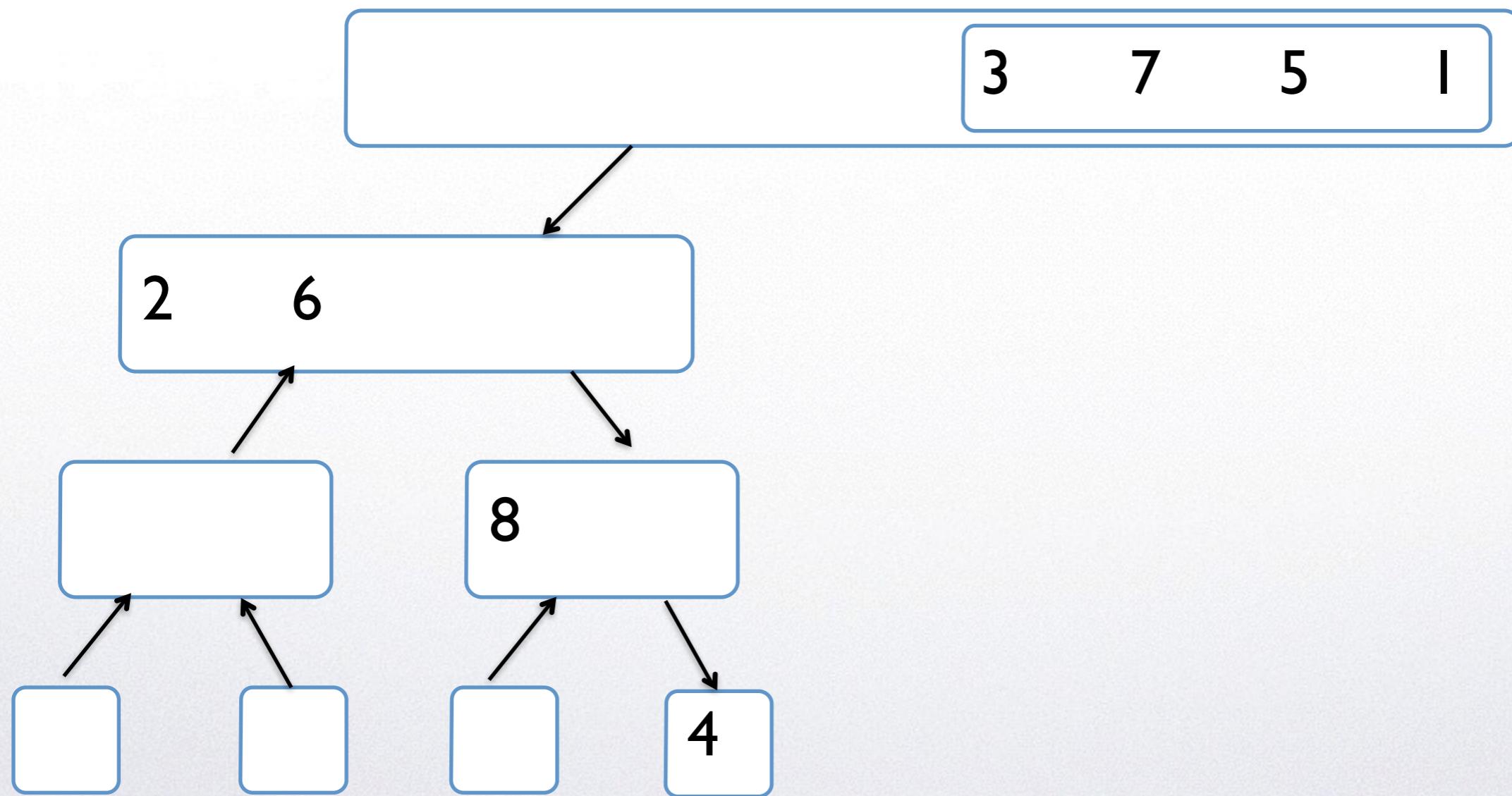


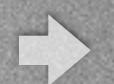
Merge (fase dividir)



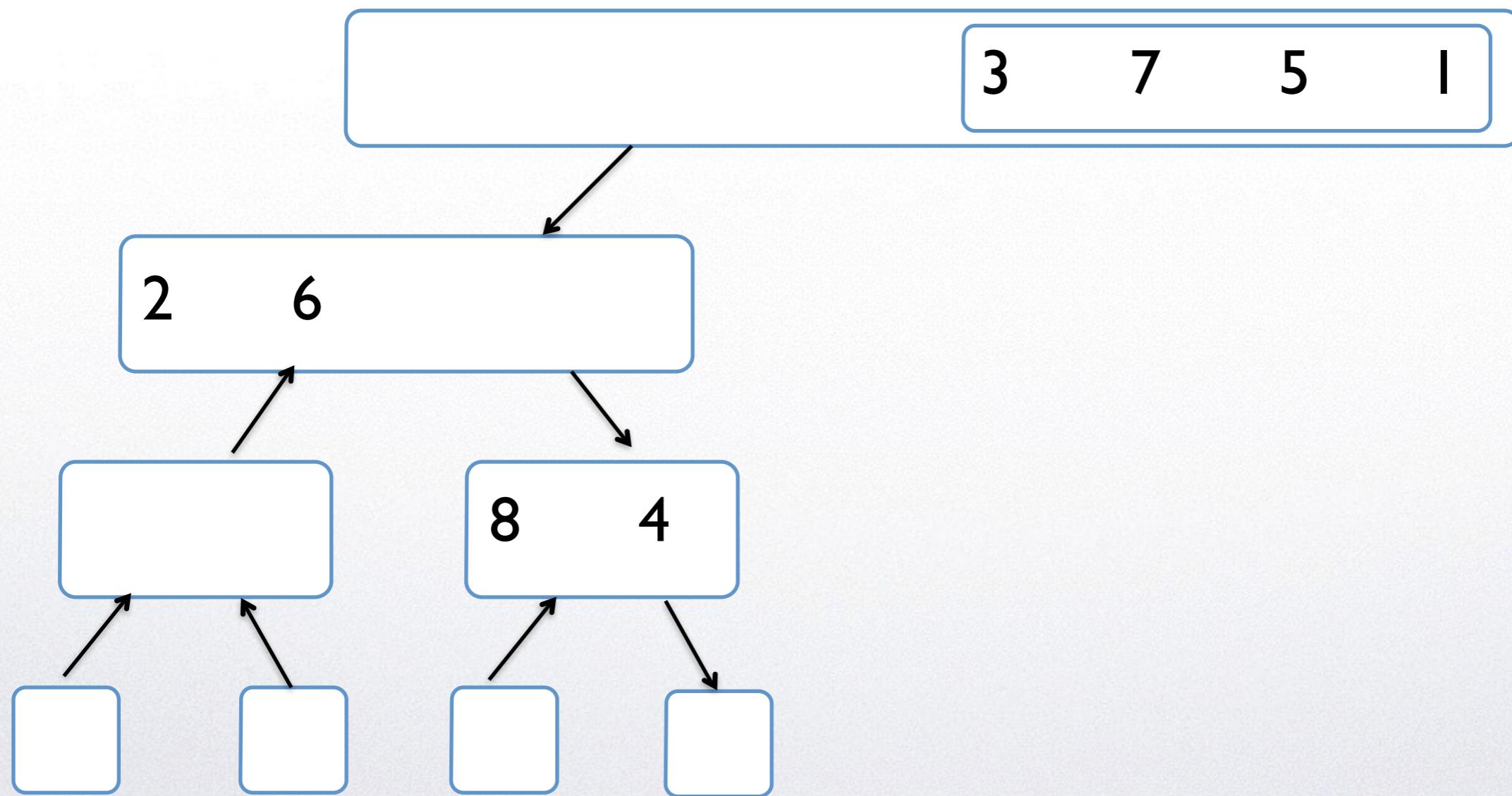


Merge (fase dividir)



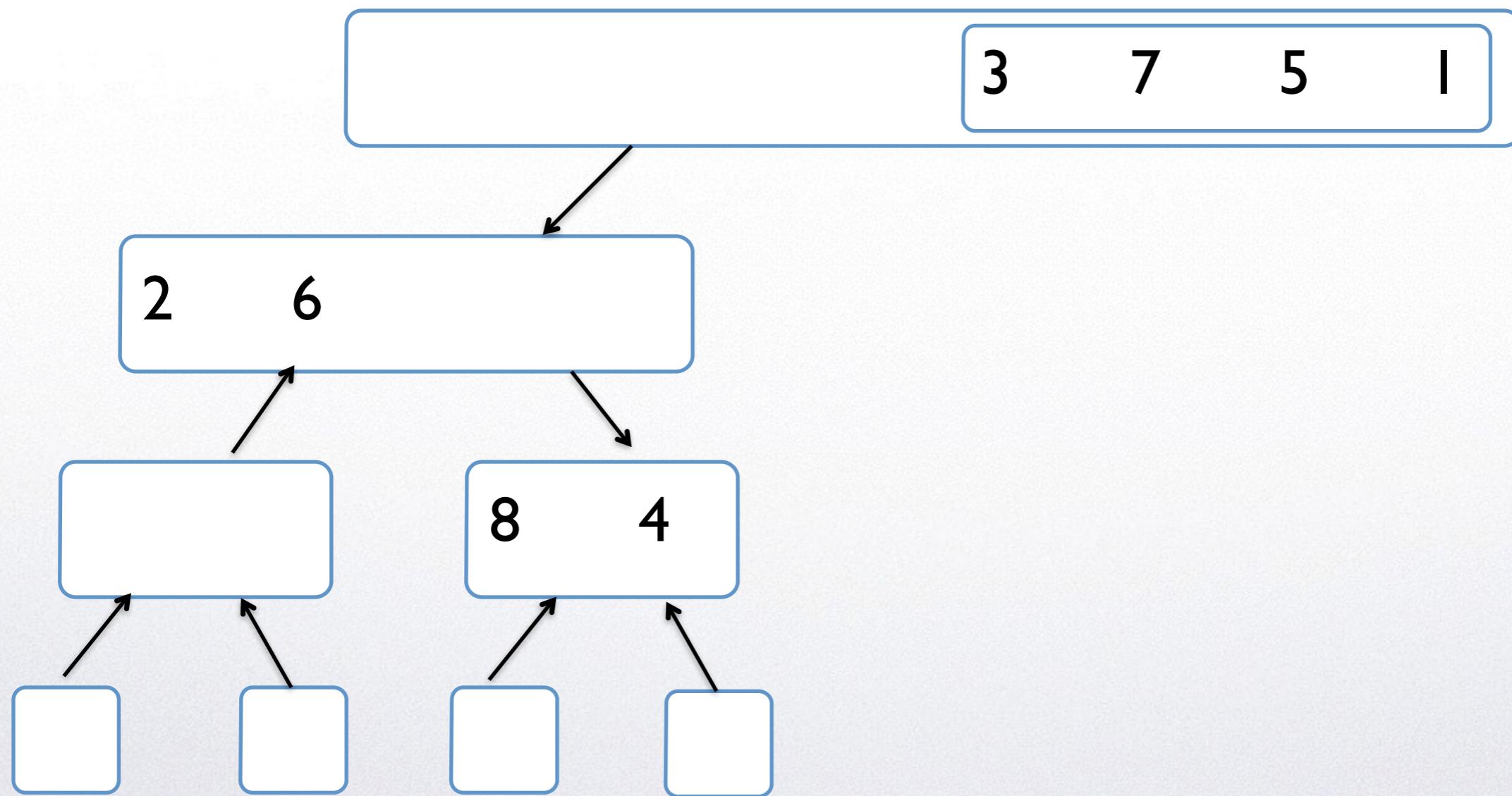


Merge (fase dividir)



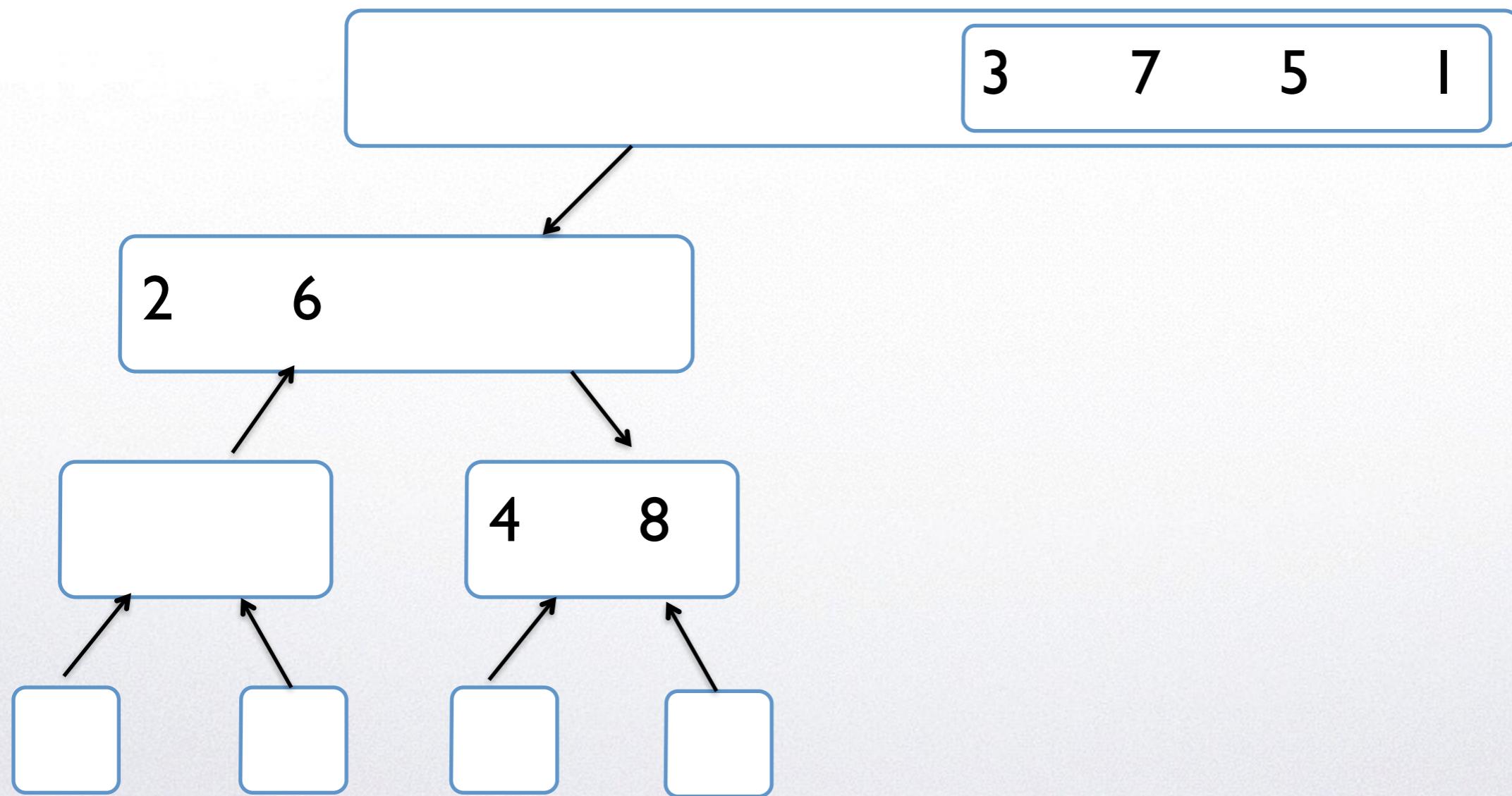


Merge (fase dividir)



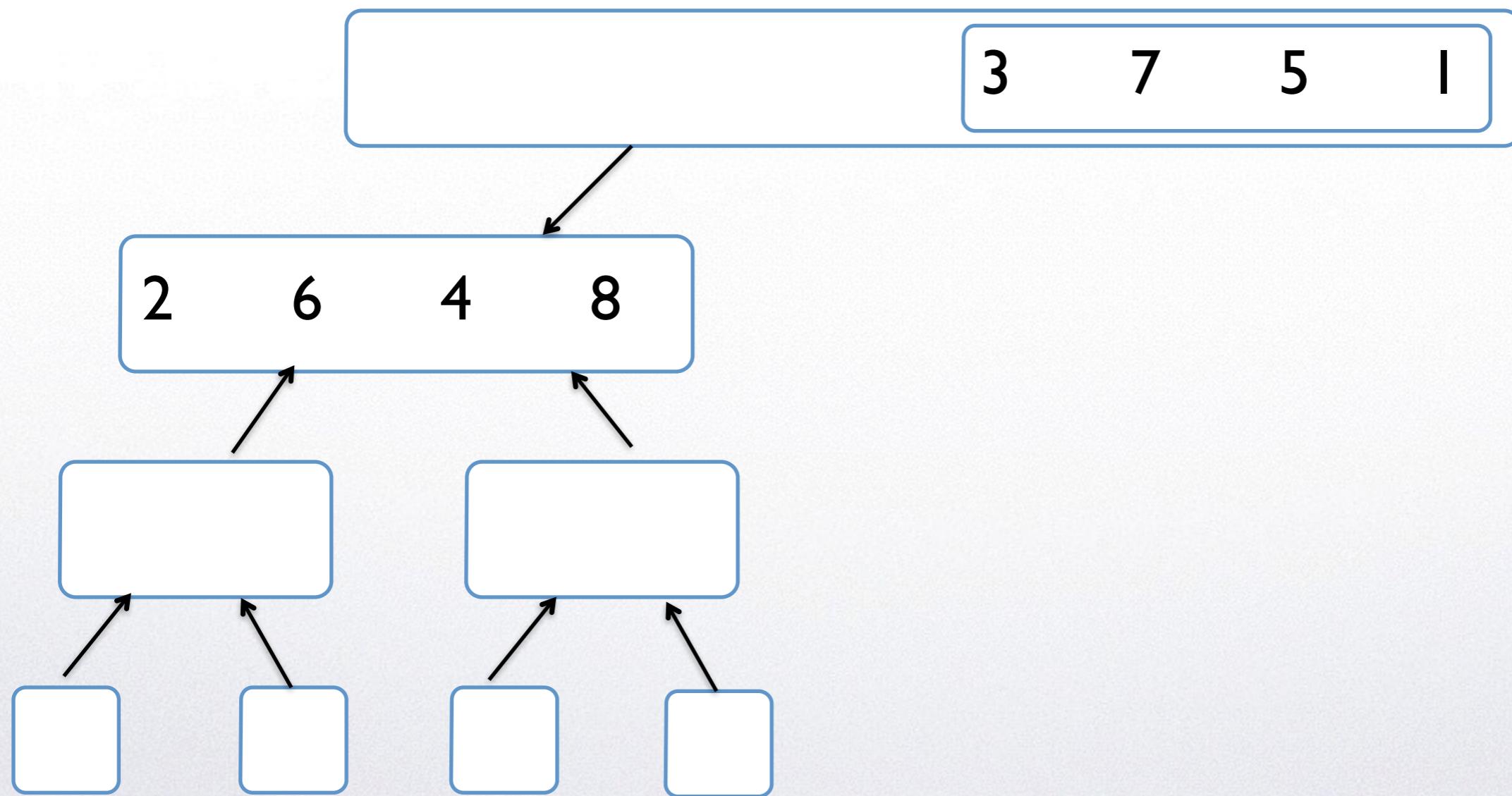


Merge (fase dividir)



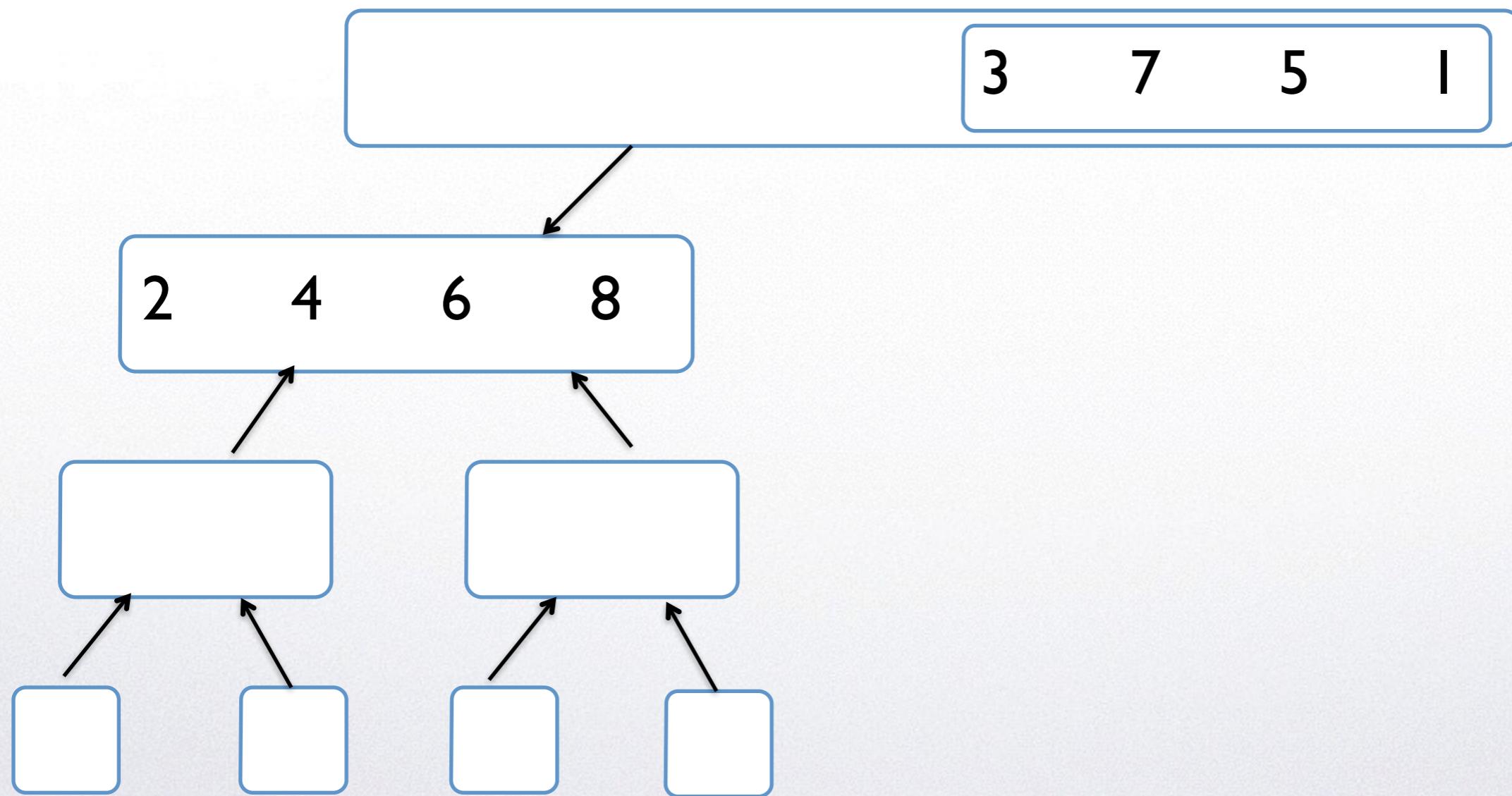


Merge (fase dividir)



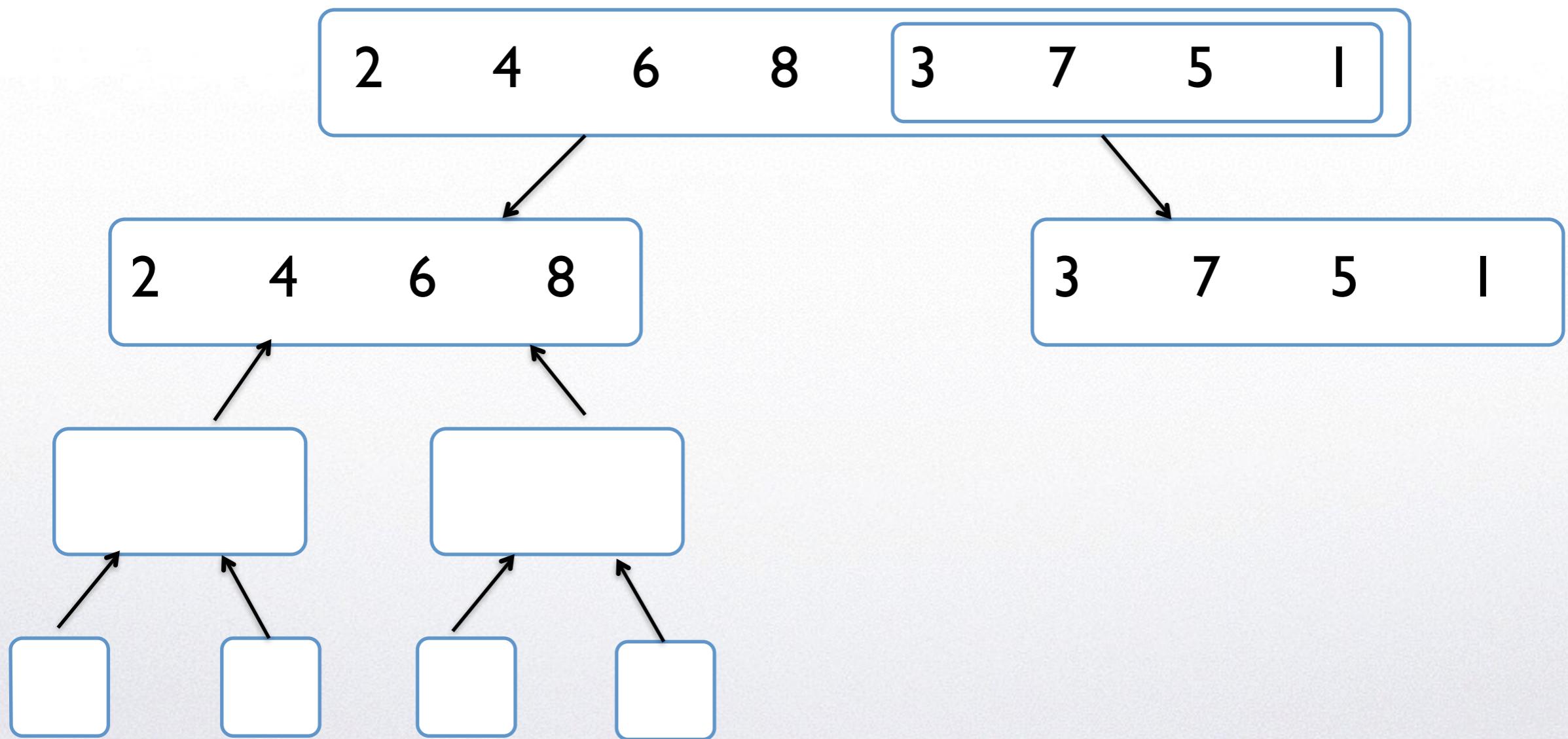


Merge (fase dividir)





Merge (fase dividir)





Revisão?



Perguntas?



"...Embora nós, seres humanos, sejamos muito limitados fisicamente, nossas mentes estão livres para explorar todo o universo..." Stephen Hawking

- O universo em uma casca de nós



Ordenação de dados

Glauco Vinicius Scheffel, Msc



"Poderia viver encerrado numa casca de noz e julgar-me rei do infinito..." Shakespeare - Hamlet, ato 2, cena 2



Dividir e Conquistar

“O problema é subdividido em um ou mais subproblemas, cada um similar ao problema original. Cada um dos subproblemas é resolvido independentemente. Finalmente, as soluções dos subproblemas são combinadas a fim de se obter a solução do problema original.”

Bruno R. Preiss



Dividir e conquistar

1. Problema pequeno, resolva o problema
2. Dividir o problema em dois ou mais problemas menores e similares
3. Recursão resolva o problema
4. Conquistar, reuna as soluções parciais unindo as seqüências

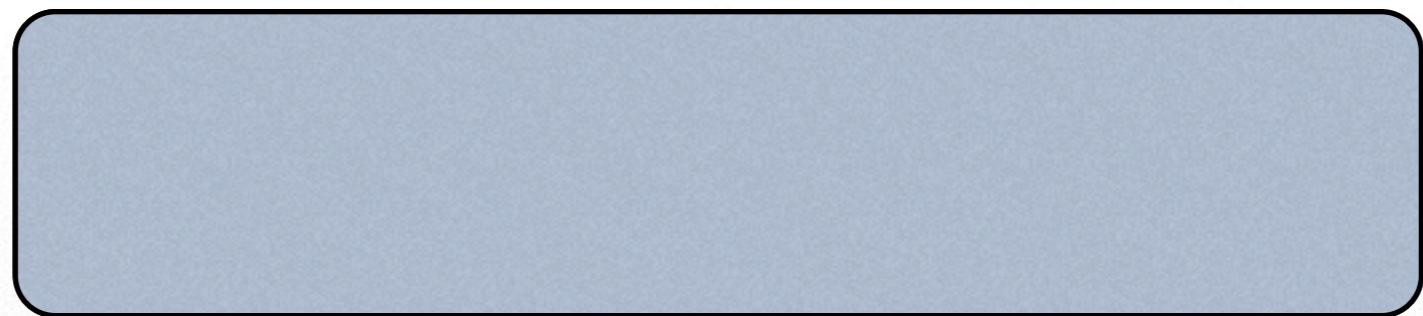


Merge Sort

Visão Geral



Visão geral Merge Sort





Visão geral Merge Sort

Parte I

Parte 2

dividir



Visão geral Merge Sort

Parte 2

Parte I

conquistar



Merge Sort
Conquista



Merge (fase conquista)

19

2

43

30

15

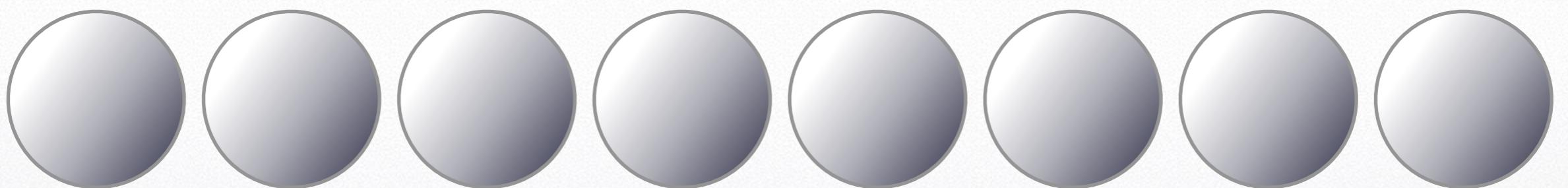
12

51

32



Merge (fase divisão)



19

2

43

30

15

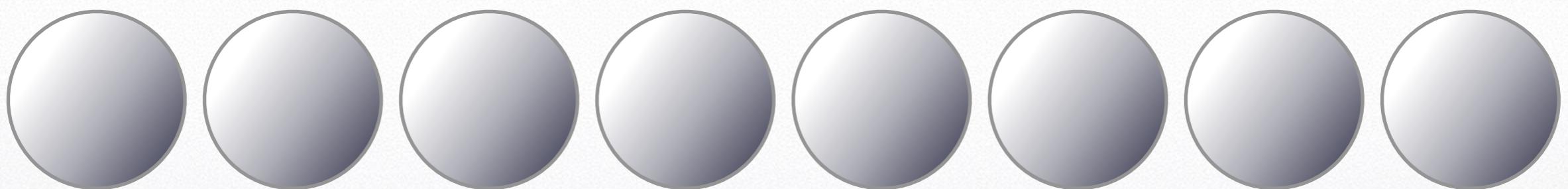
12

51

32



Merge (fase divisão - ordenar)



2

19

30

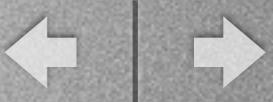
43

15

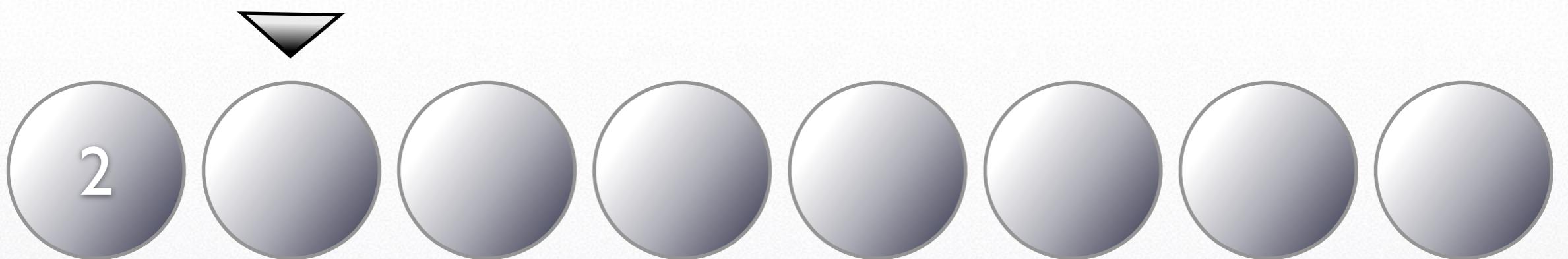
12

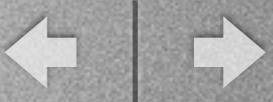
51

32

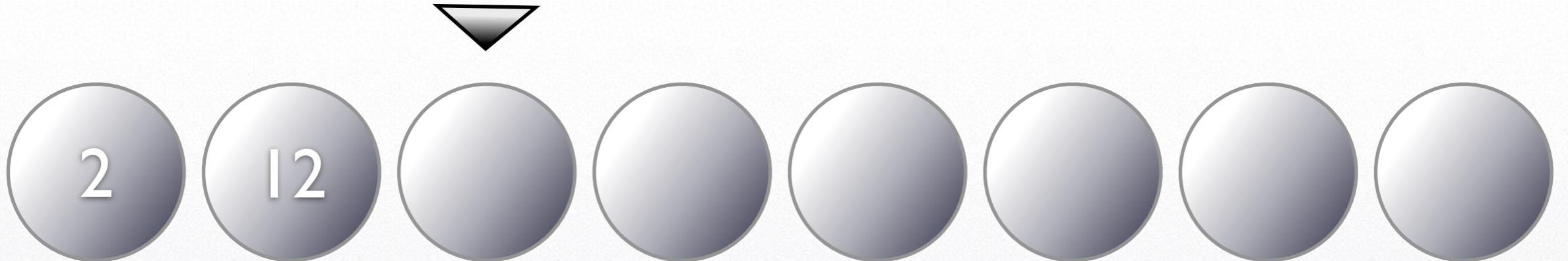


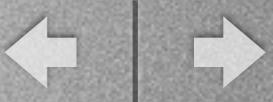
Merge (fase conquista)





Merge (fase conquista)





Merge (fase conquista)

2

12

15

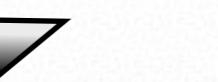


2

19

30

43



12

15

32

51





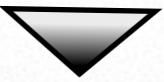
Merge (fase conquista)

2

12

15

19



2

19

30

43

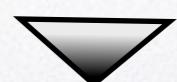


12

15

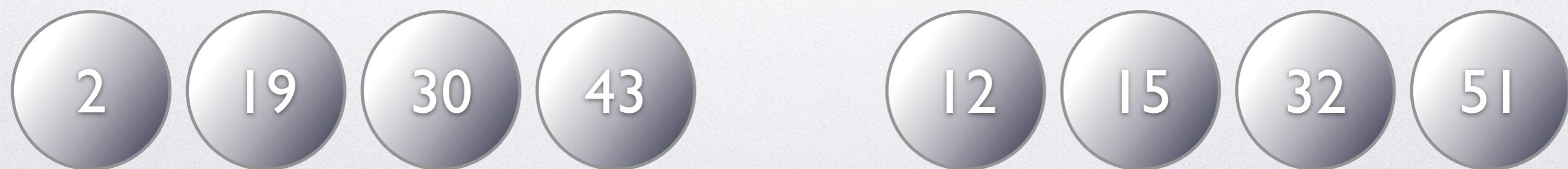
32

51



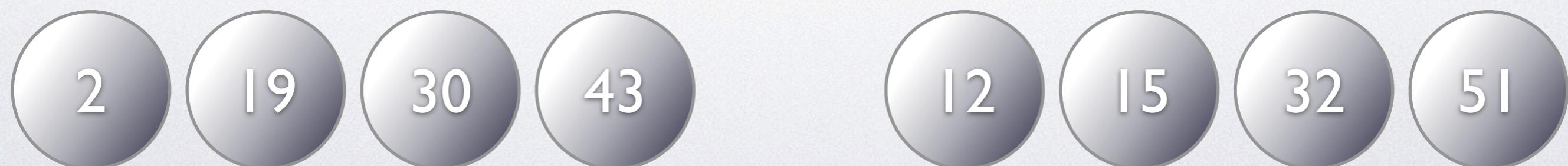
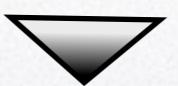


Merge (fase conquista)





Merge (fase conquista)





Merge (fase conquista)

2

12

15

19

30

32

43



2

19

30

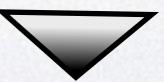
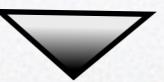
43

12

15

32

51





Merge (fase conquista)

2

12

15

19

30

32

43

51



2

19

30

43

12

15

32

51



Merge Sort Algoritmo



Merge Sort: Algoritmo

Merge-Sort (array, esquerda, direita)

se esquerda \geq direita então

retornar

senão

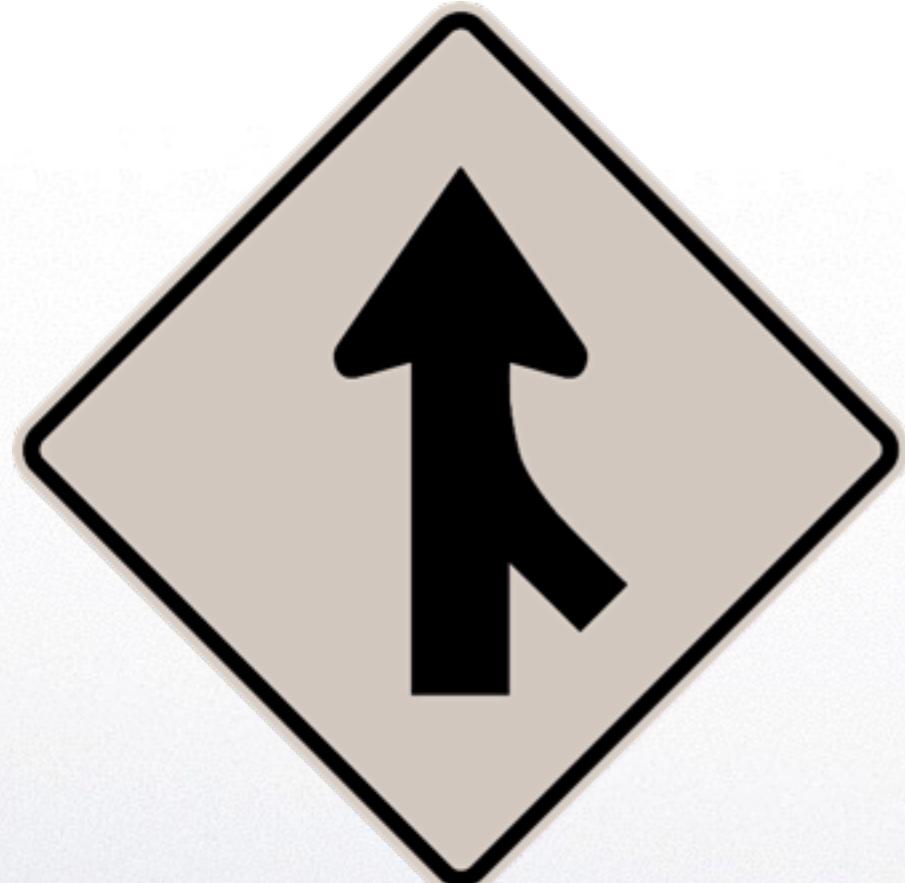
area = esquerda+direita

meio = arredondar(area / 2)

Merge-Sort(array, esquerda, meio)

Merge-Sort(array, meio+1, direita)

Merge(array, esquerda, meio, direita)





Merge: Algoritmo



função **Merge**(A, esquerda, meio, direita)

$n1 = \text{meio} - \text{esquerda} + 1$

$n2 = \text{direita} - \text{meio}$

criar array temporario L[$n1$], R[$n2$]

for $i = 0$ to $n1-1$ faça

$L[i] = A[\text{esquerda} + i]$

for $j = 0$ to $n2-1$ faça

$R[j] = A[\text{meio}+j]$

zerar variáveis contadoras do loop



Merge: Algoritmo

enquanto $i < n1 \ \&\& \ j < n2$ faça

 se $L[i] < R[j]$ então

 array[k++] = ESQUERDA[i++]

 senão

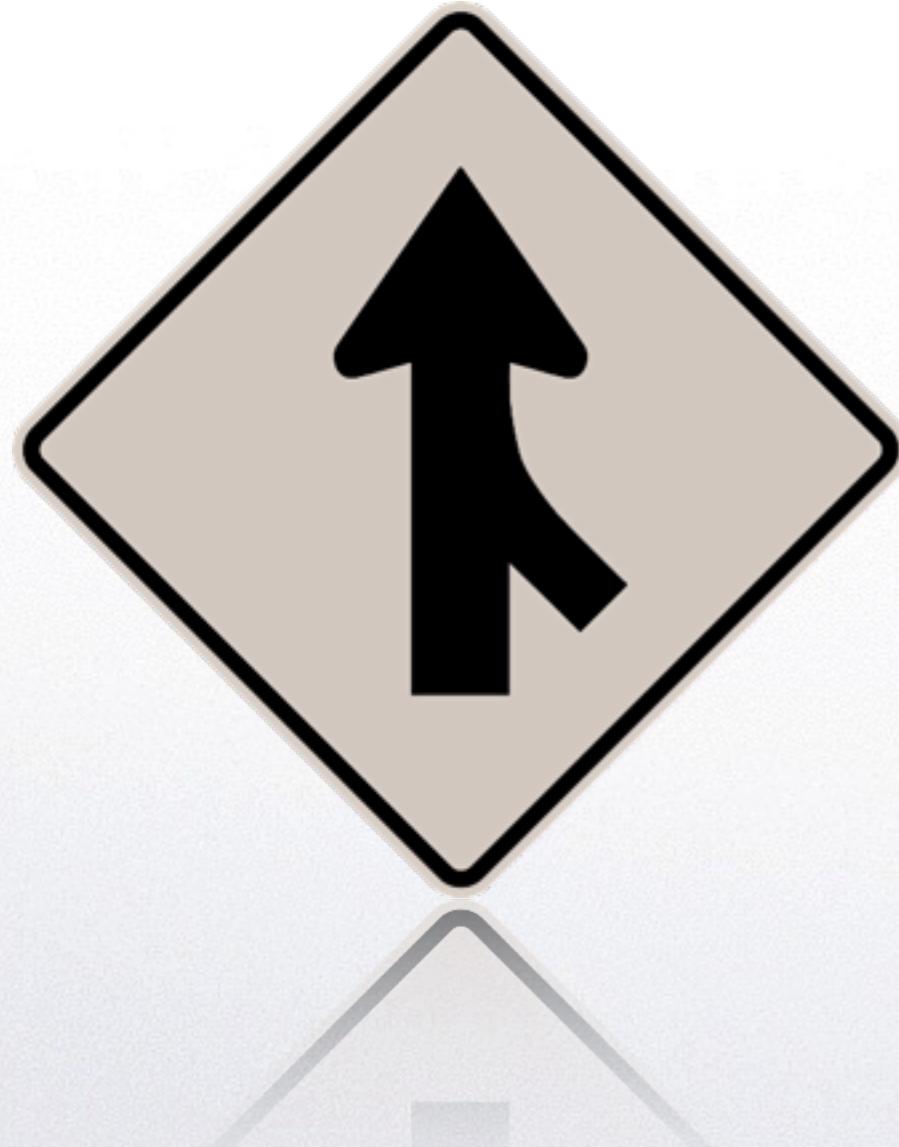
 array[k++] = DIREITA[j++]

enquanto $i < n1$ faça

 direita[k++] = ESQUERDA[i++]

enquanto $j < n2$ faça

 array[k++] = DIREITA[j++]





Merge Sort

Divisão

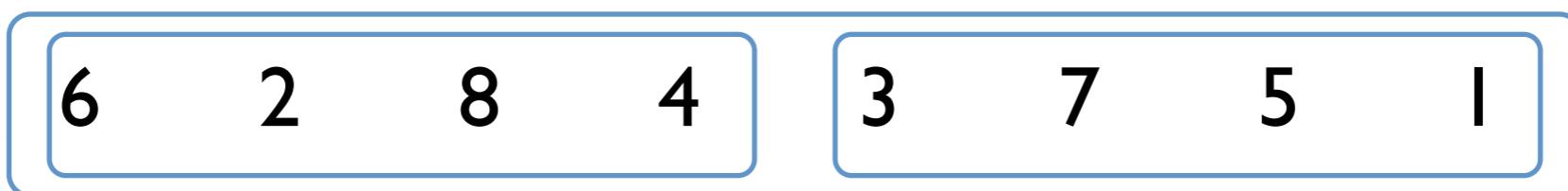


Merge (fase dividir)

6 2 8 4 3 7 5 |



Merge (fase dividir)



Merge-Sort(A, 0, 7),

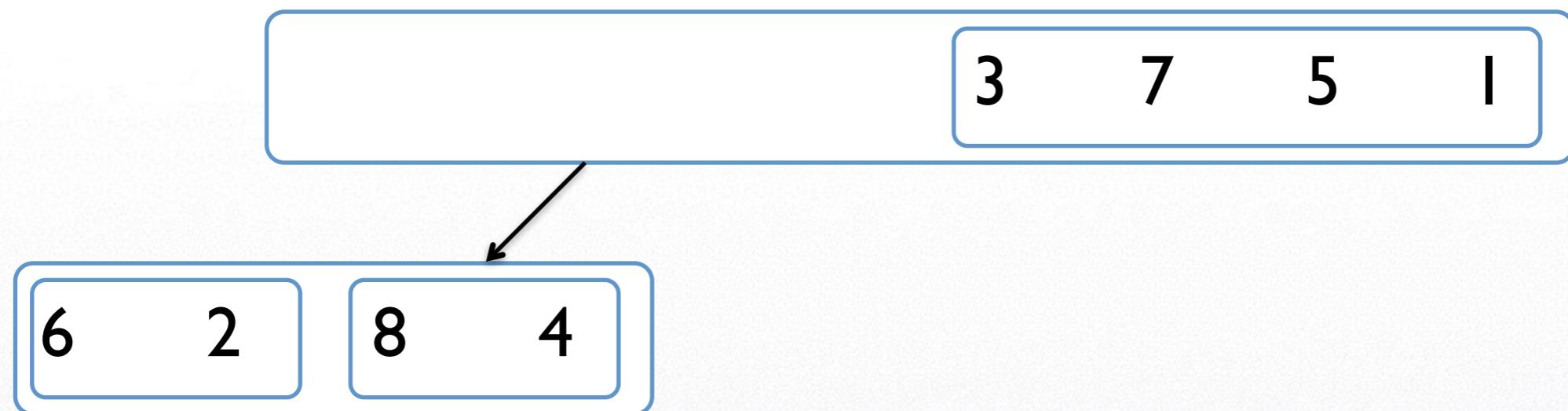


Merge (fase dividir)





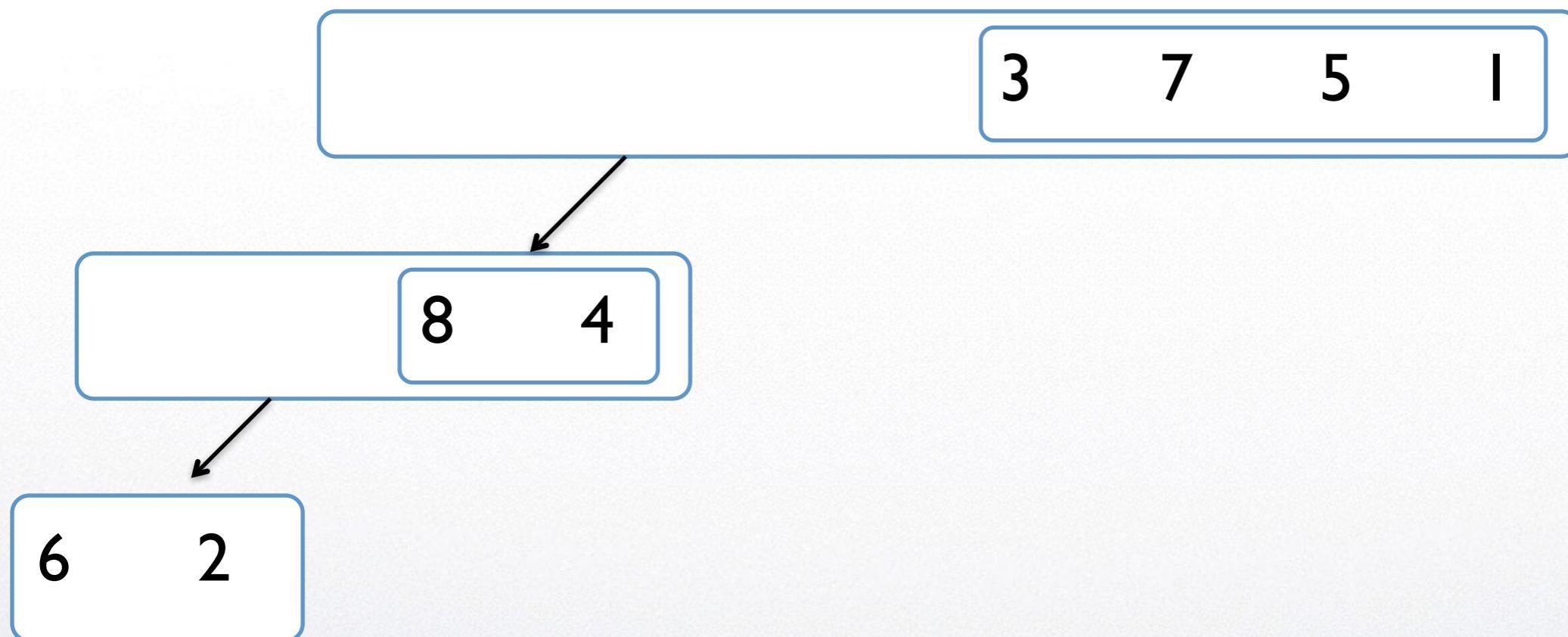
Merge (fase dividir)



Merge-Sort(A, 0, 7), Merge-Sort(A,0,3) dividir



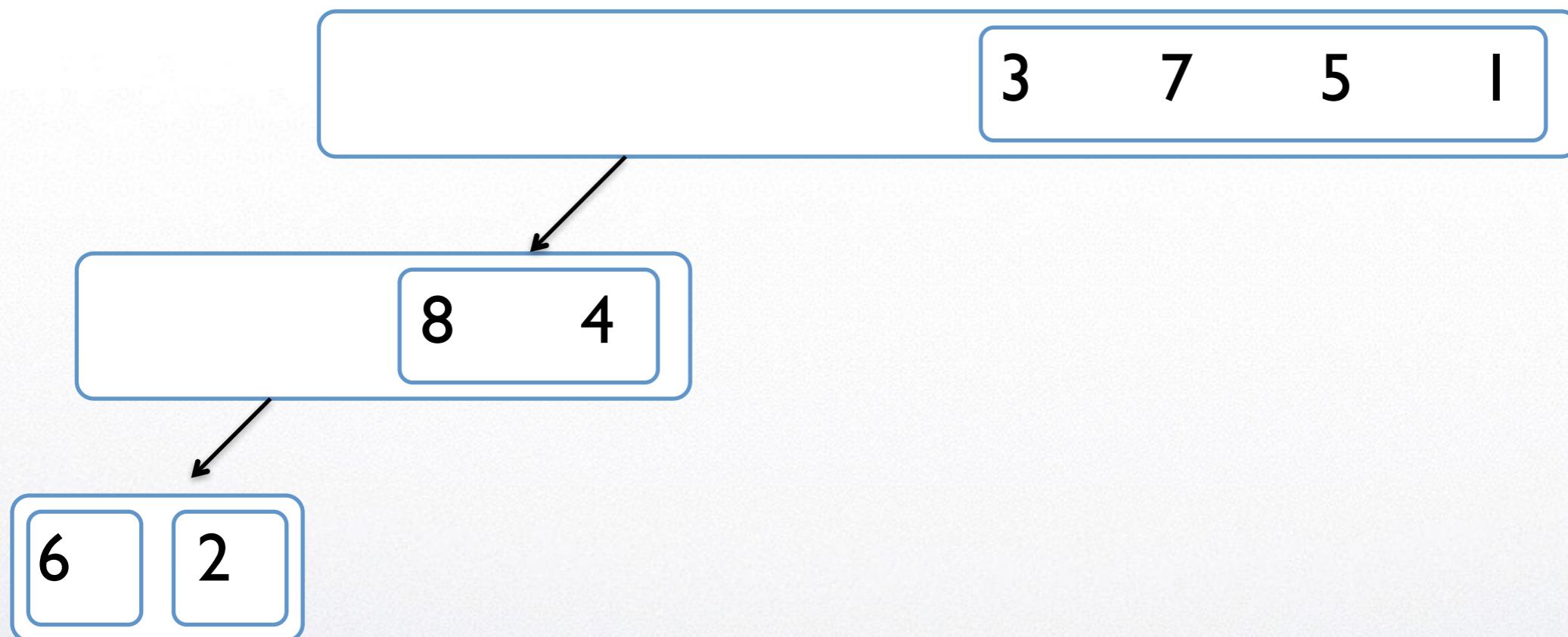
Merge (fase dividir)



Merge-Sort(A, 0, 7), Merge-Sort(A,0,1) dividir

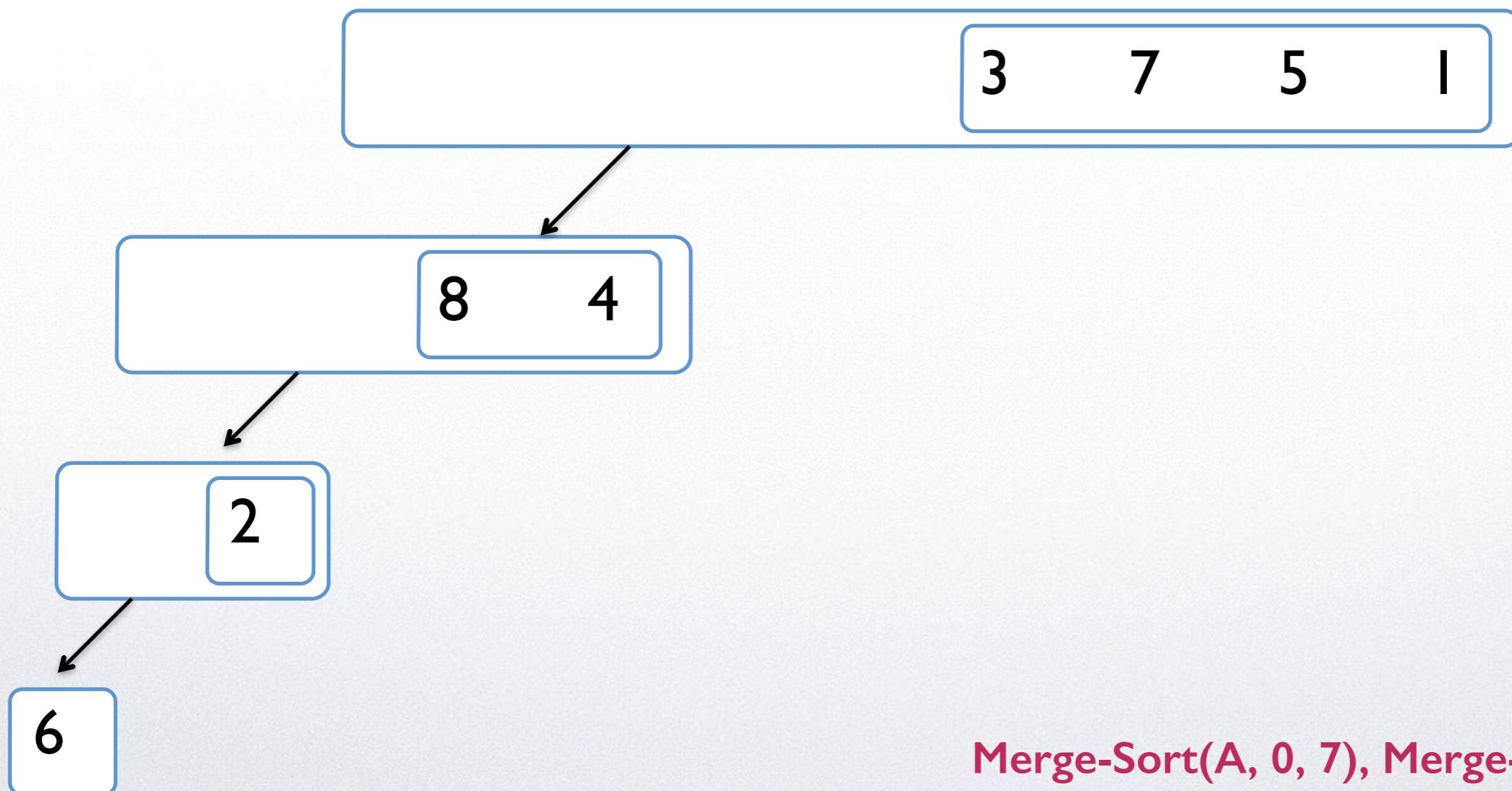


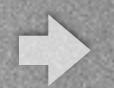
Merge (fase dividir)



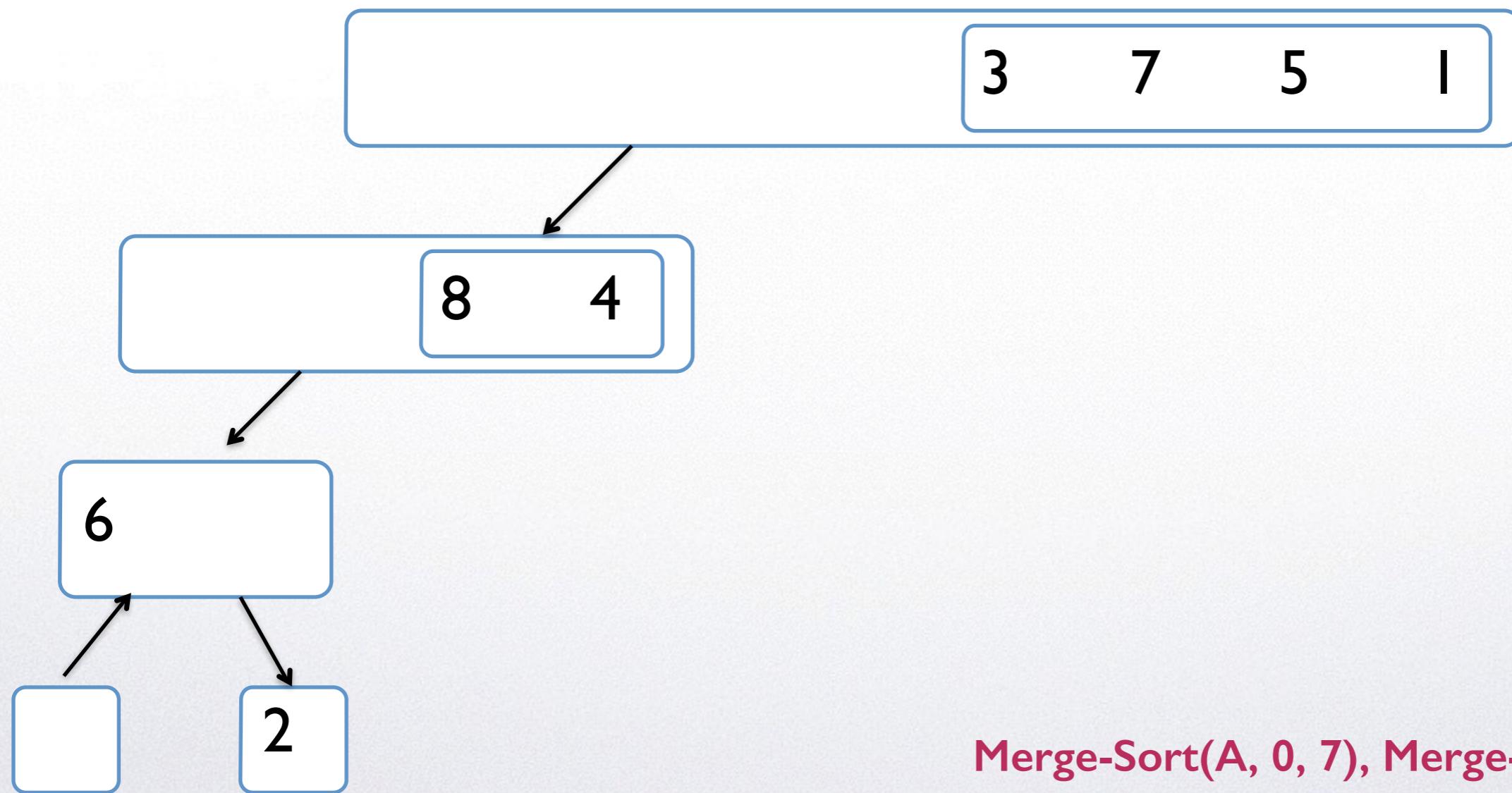


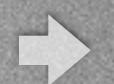
Merge (fase dividir)



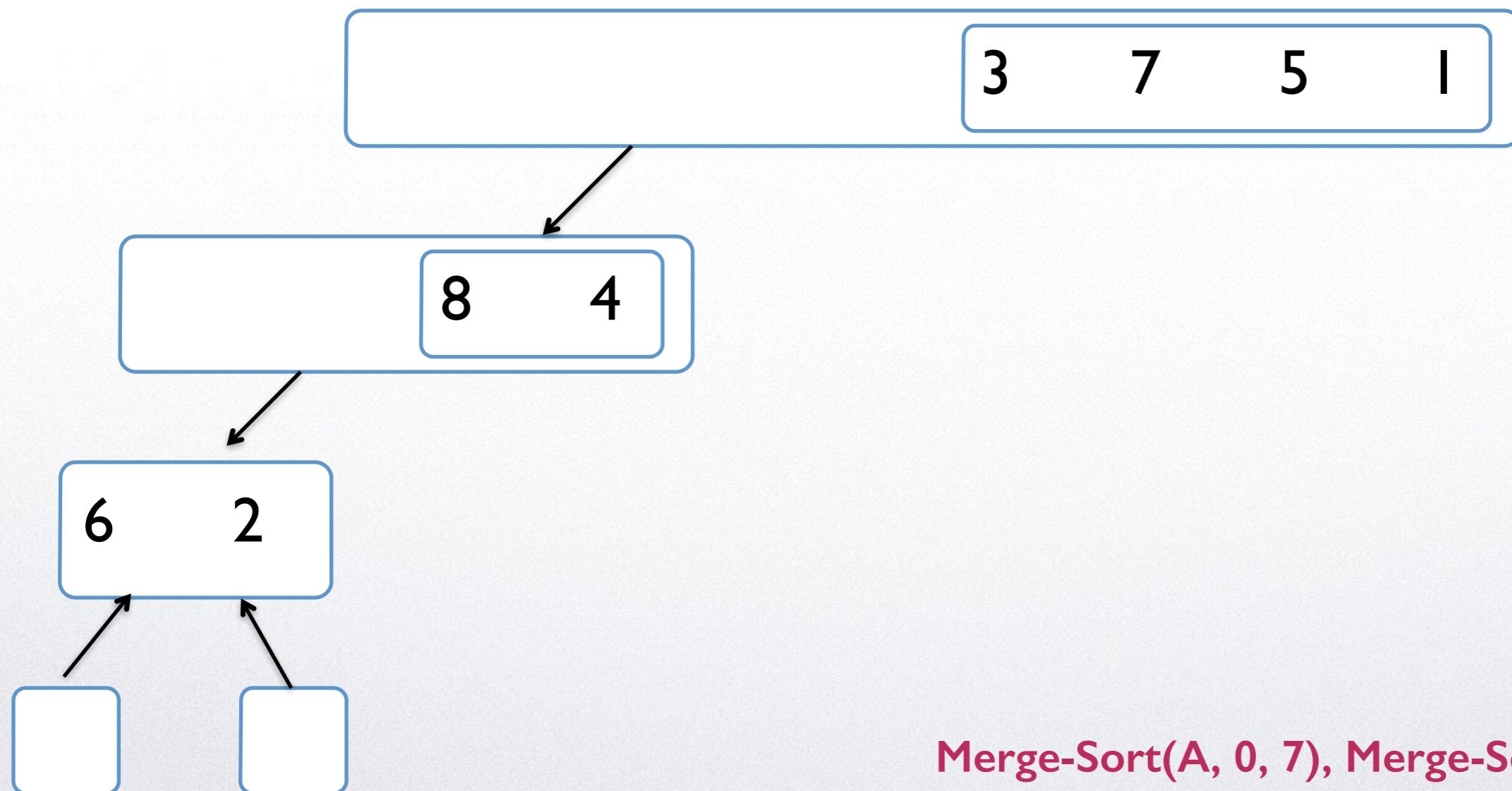


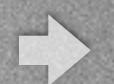
Merge (fase dividir)



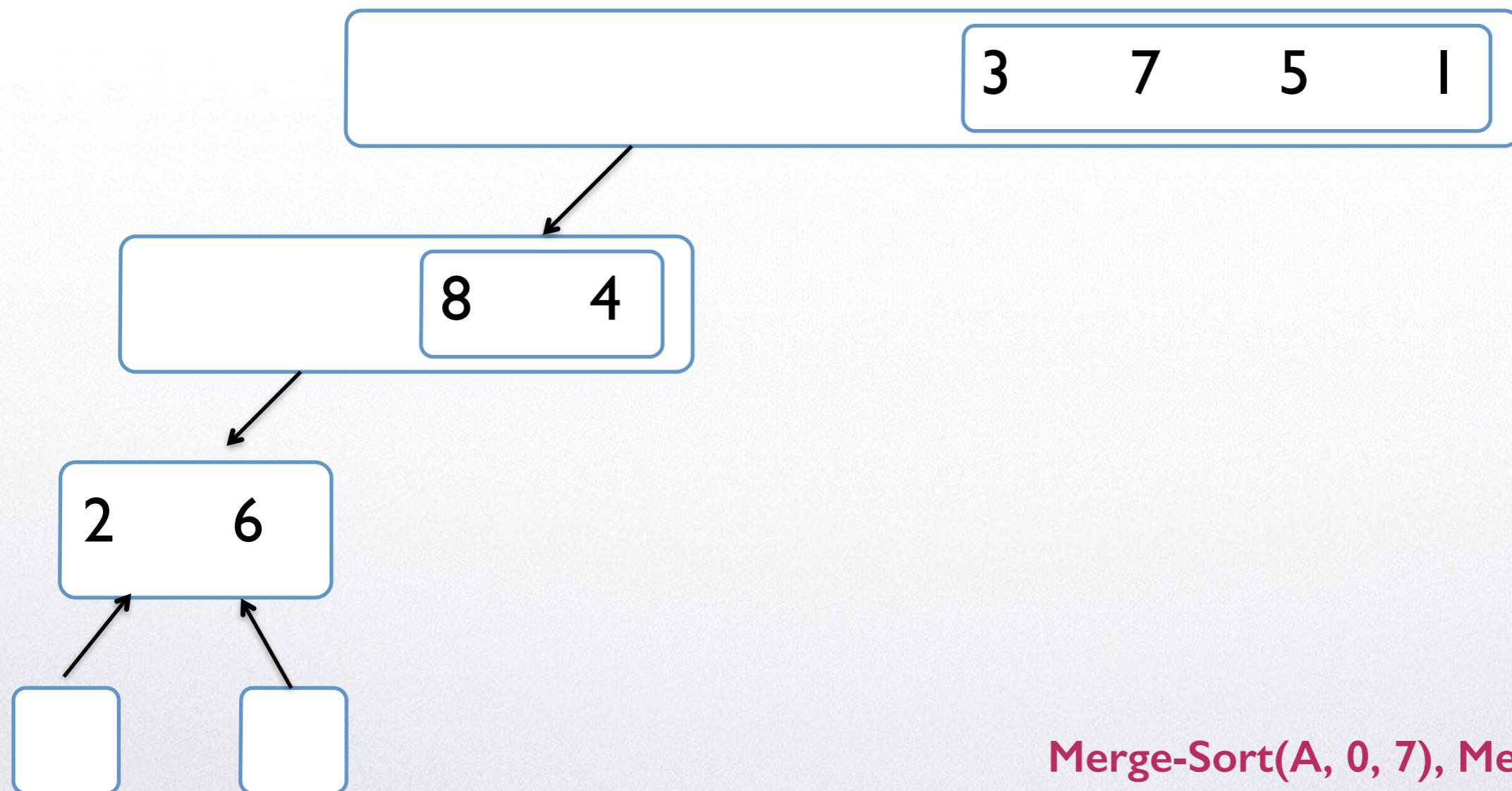


Merge (fase dividir)



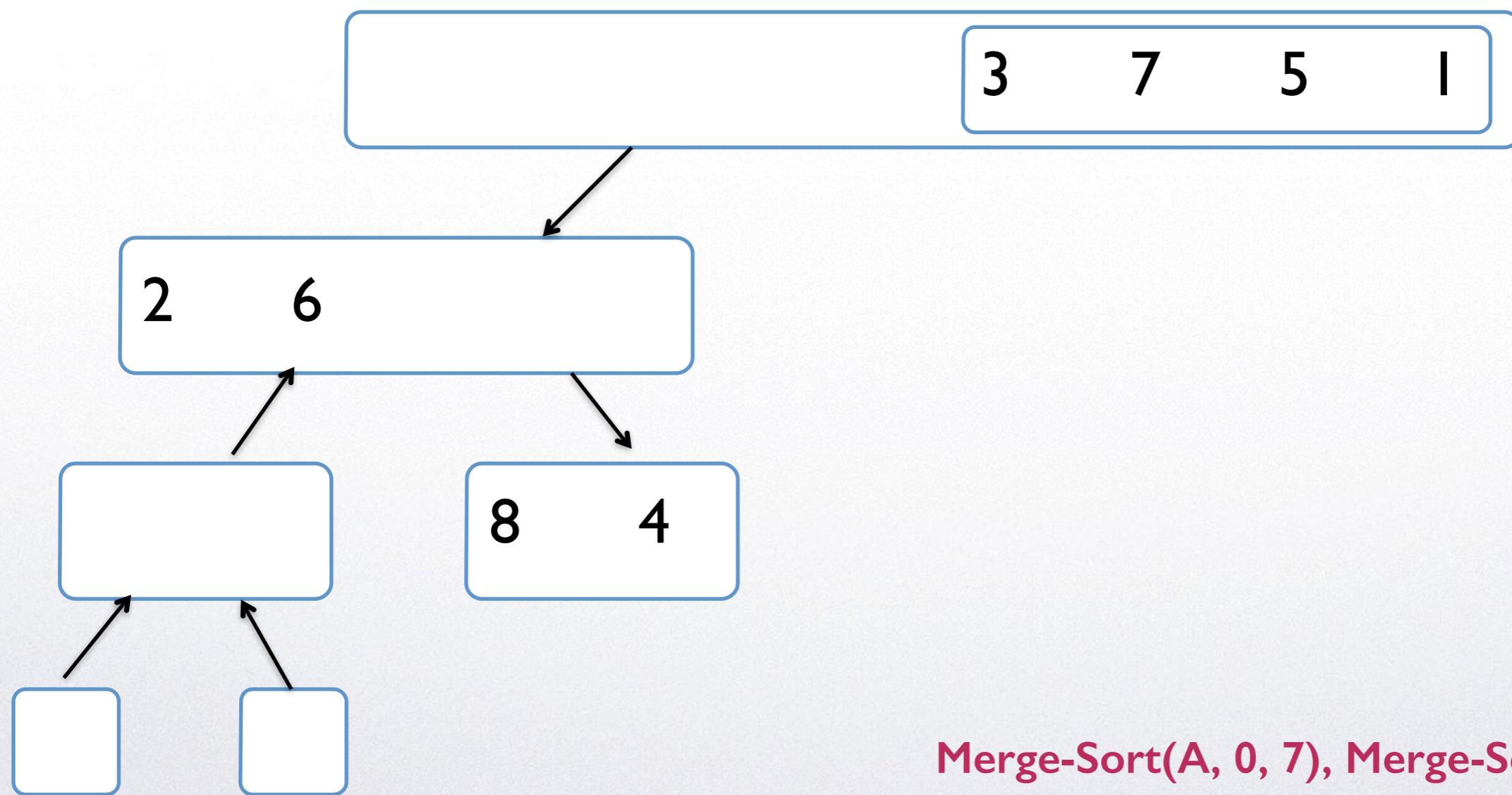


Merge (fase dividir)



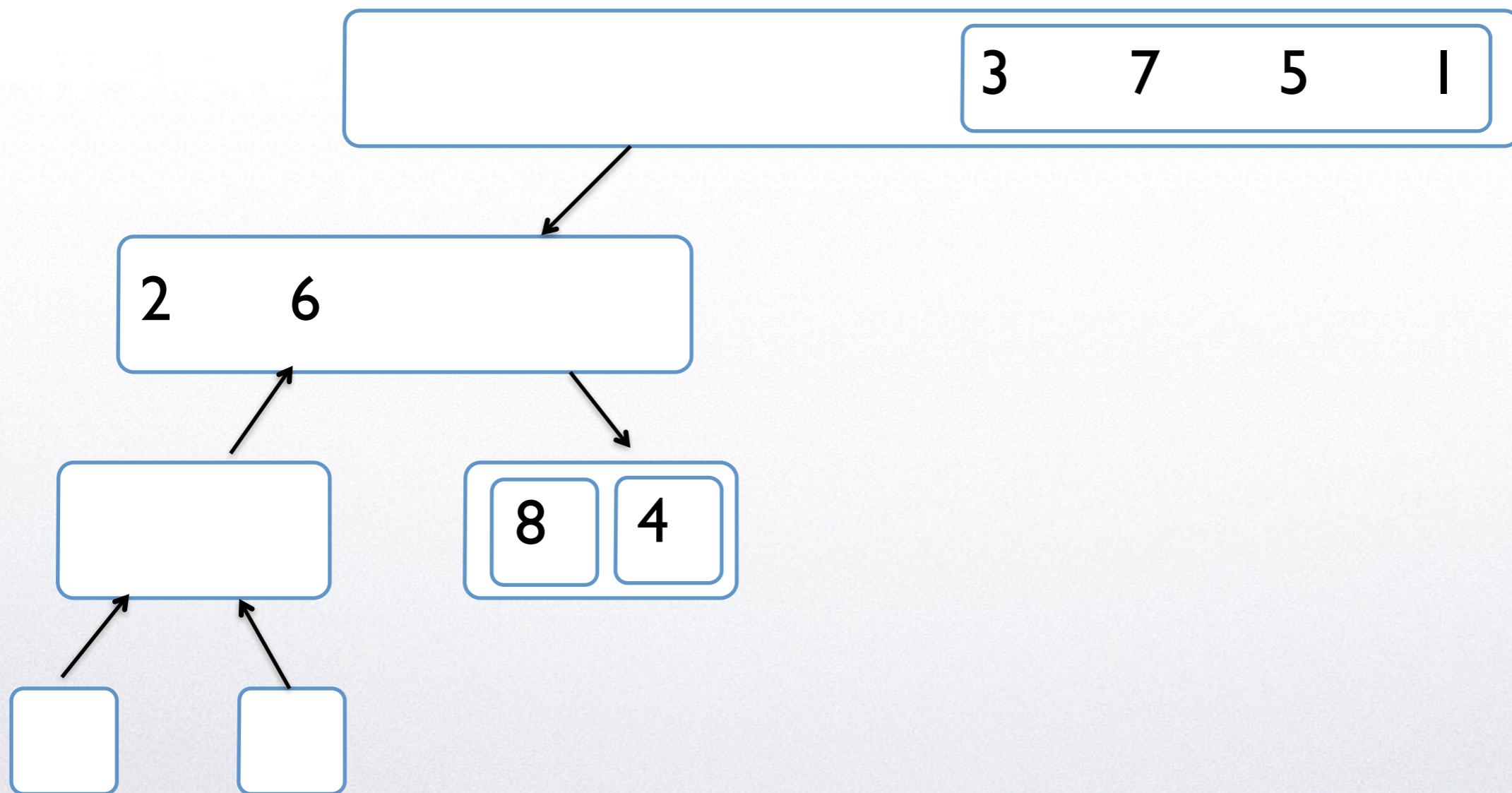


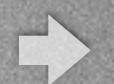
Merge (fase dividir)



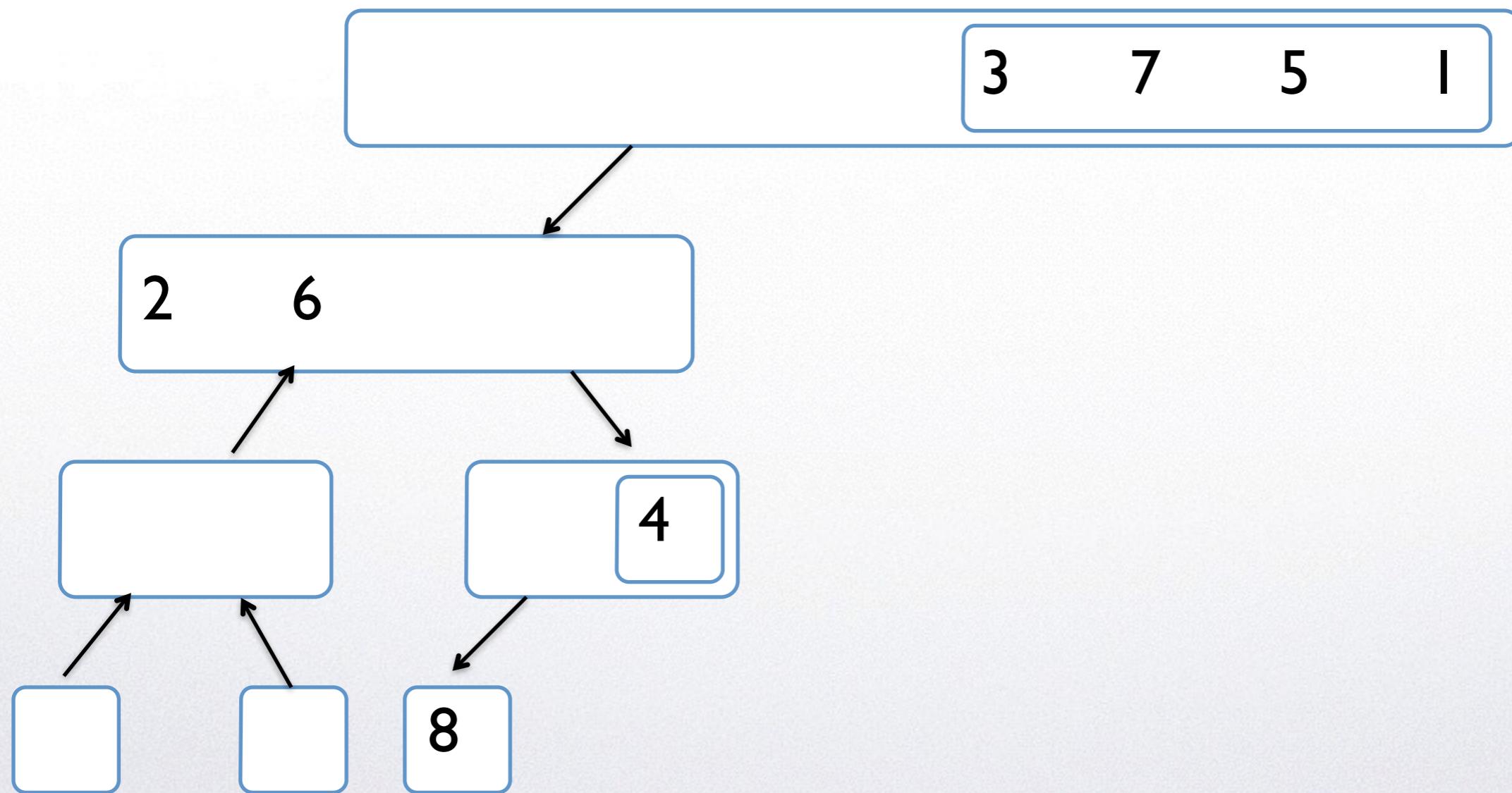


Merge (fase dividir)



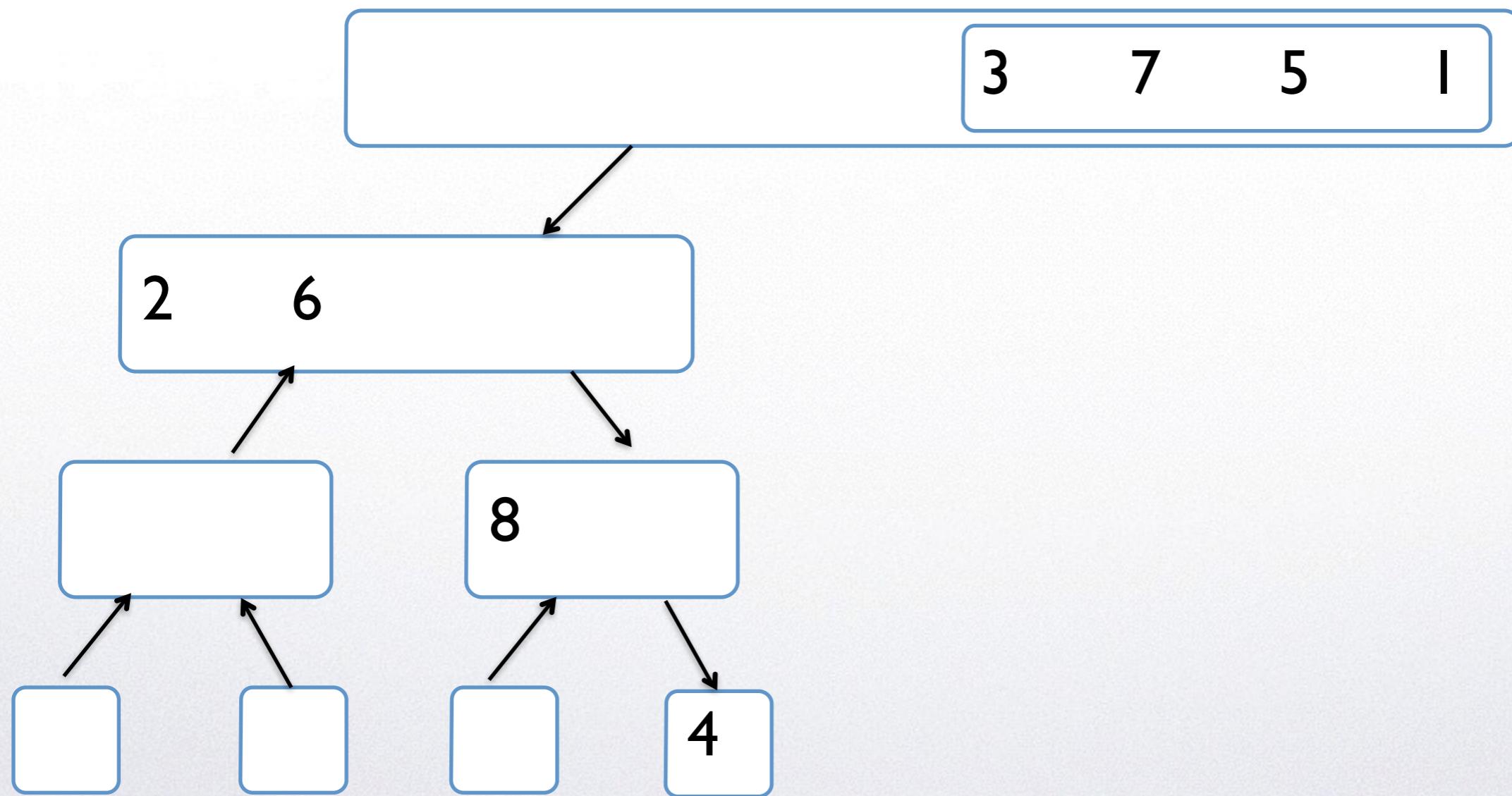


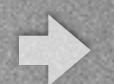
Merge (fase dividir)



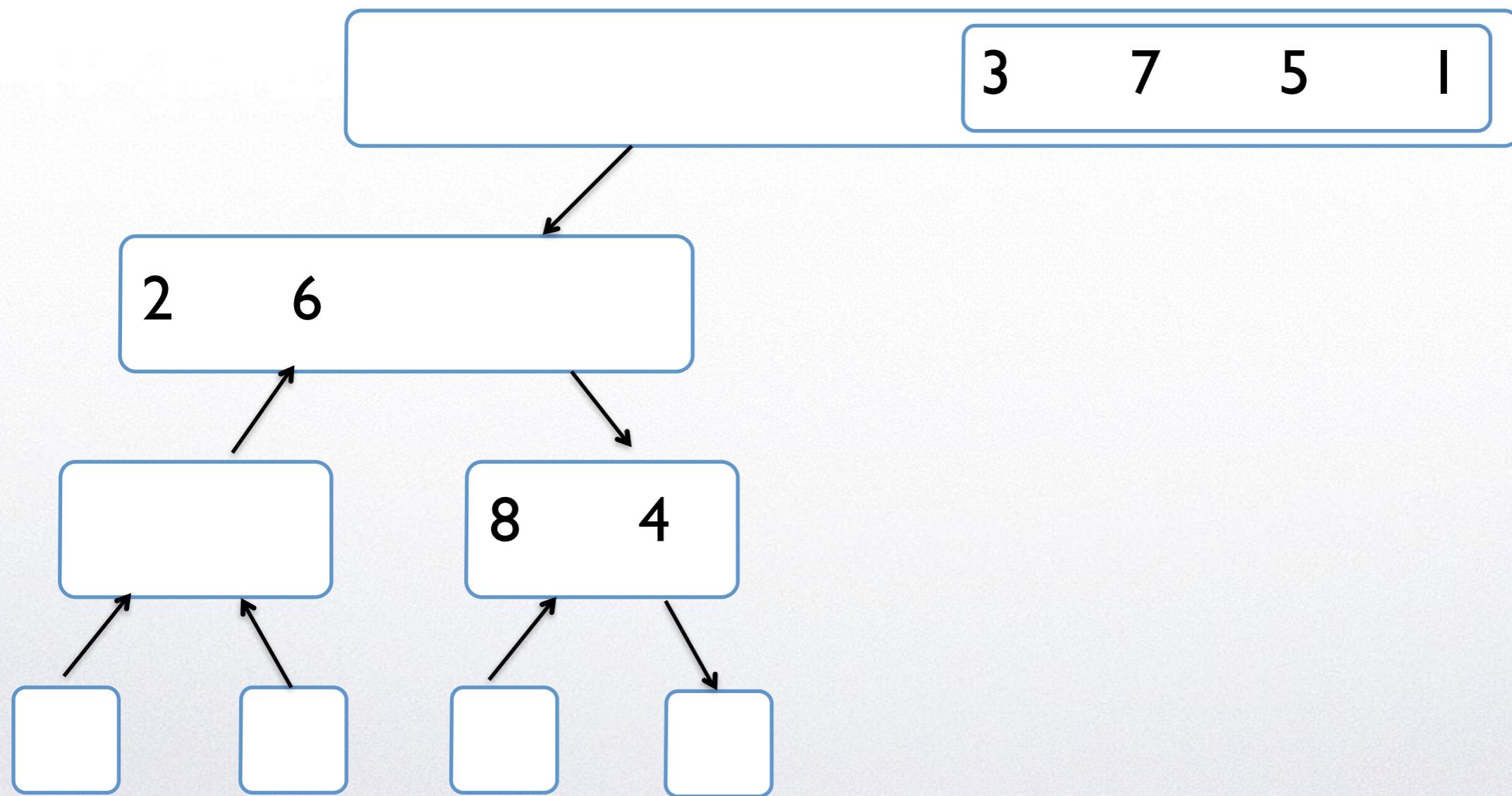


Merge (fase dividir)



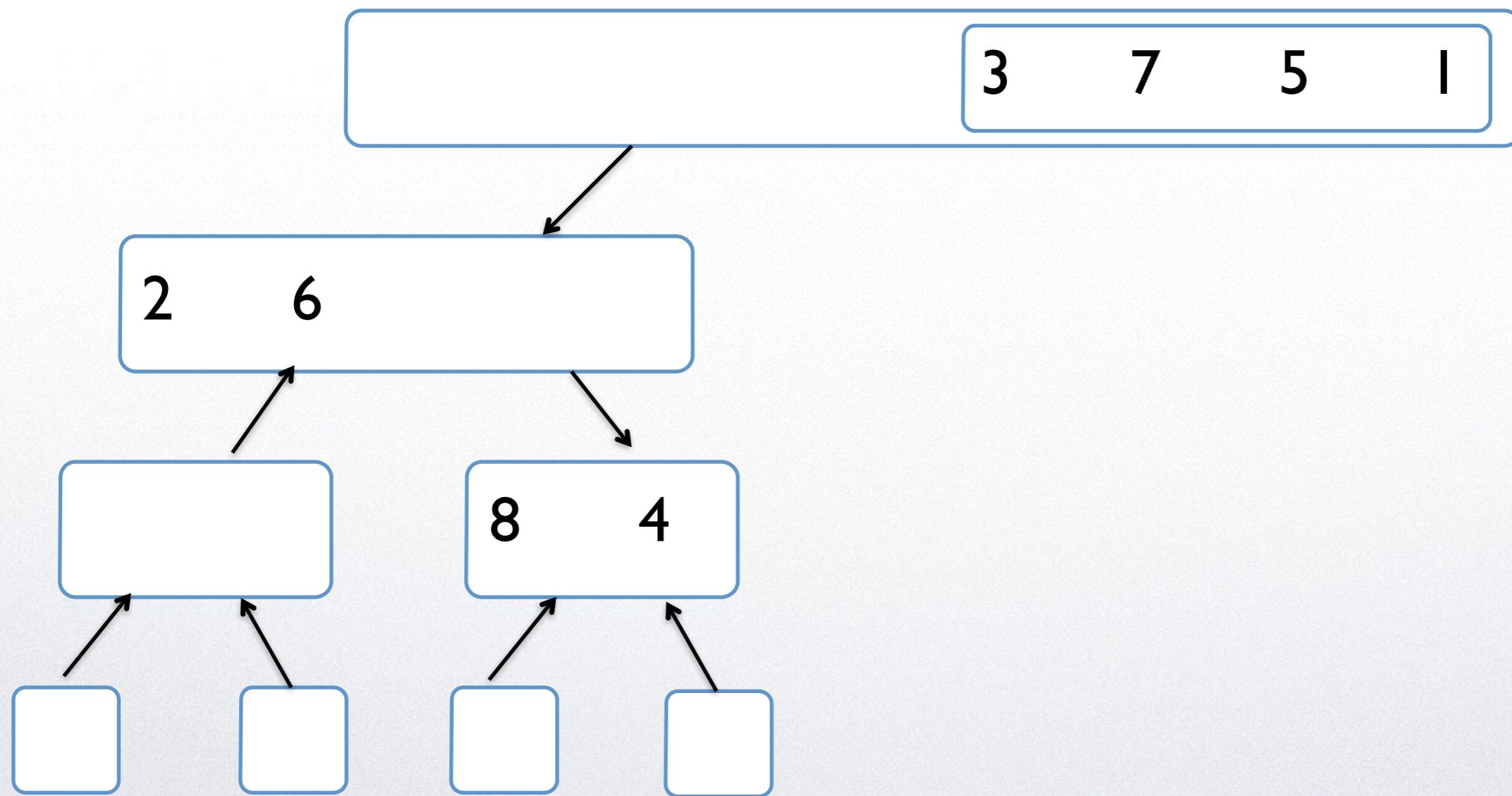


Merge (fase dividir)



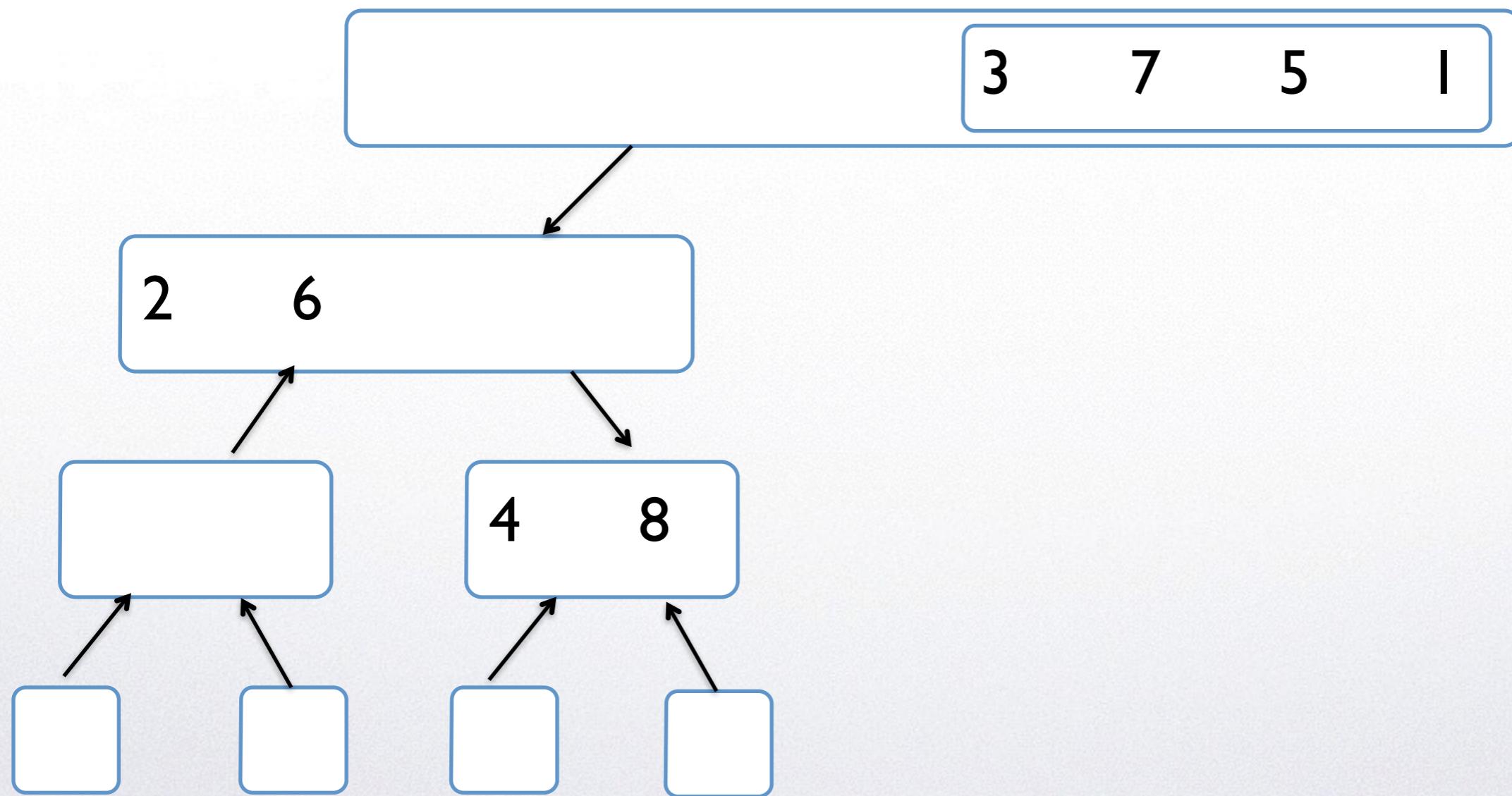


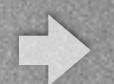
Merge (fase dividir)



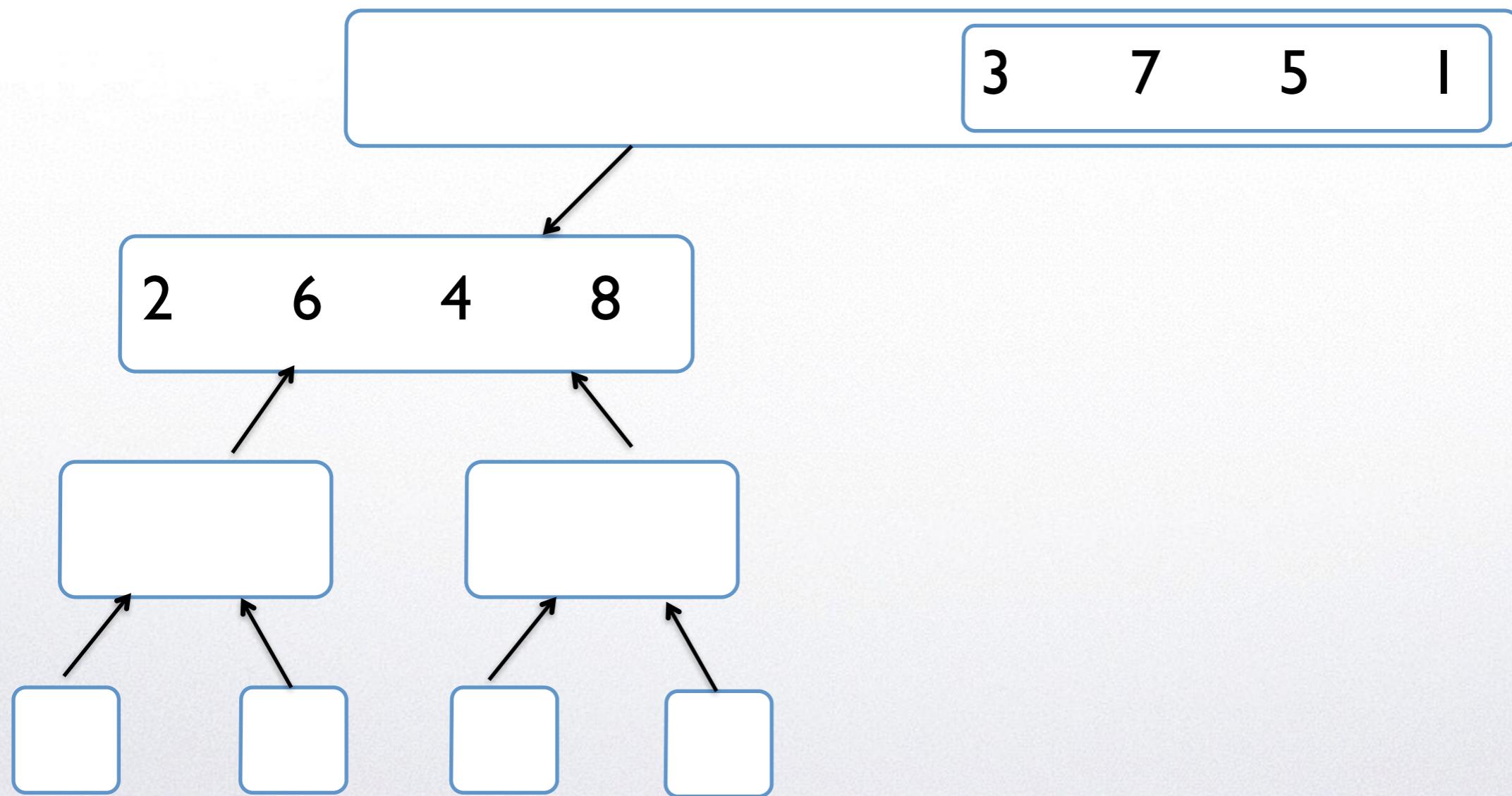


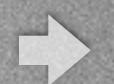
Merge (fase dividir)



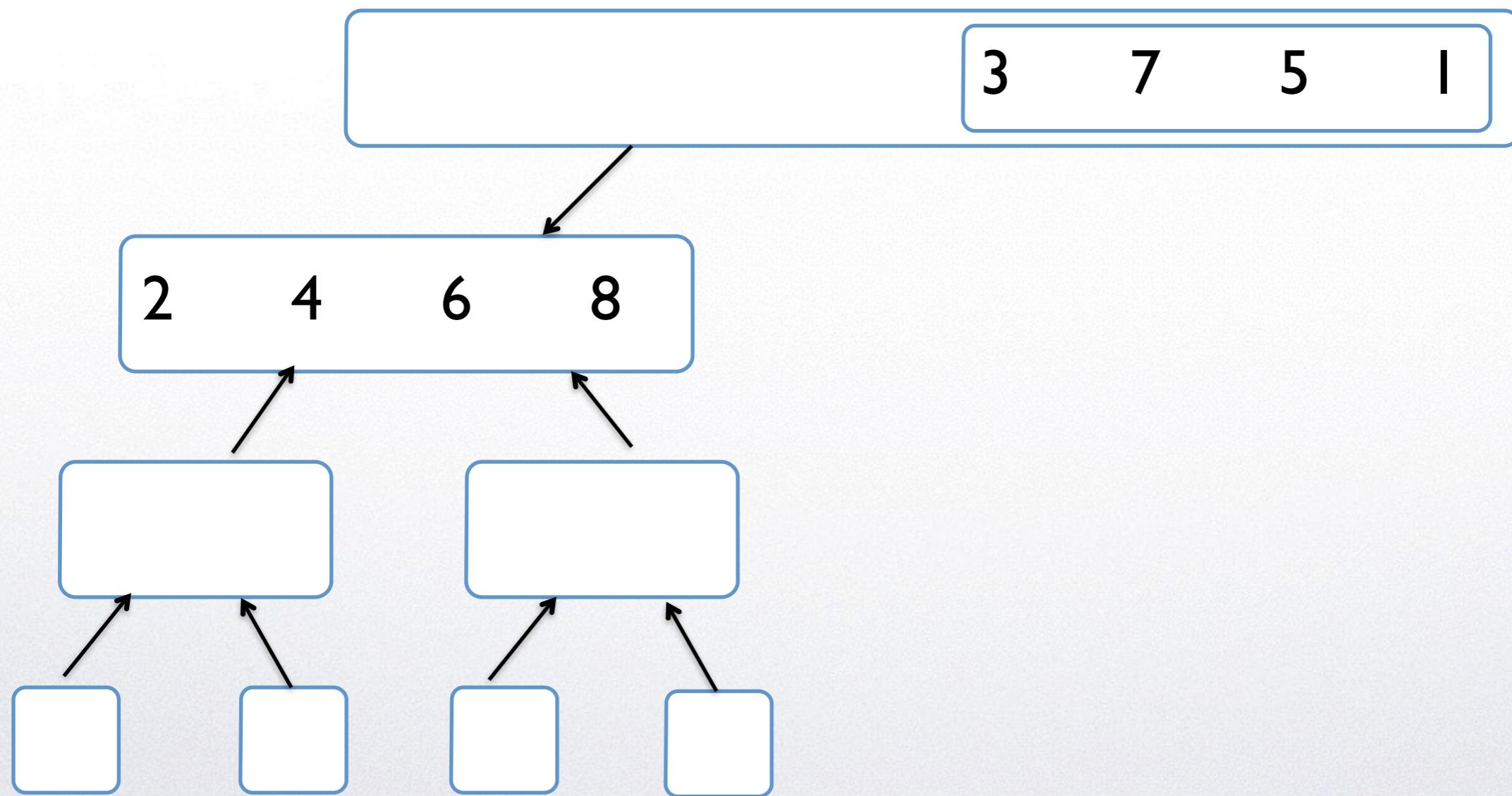


Merge (fase dividir)



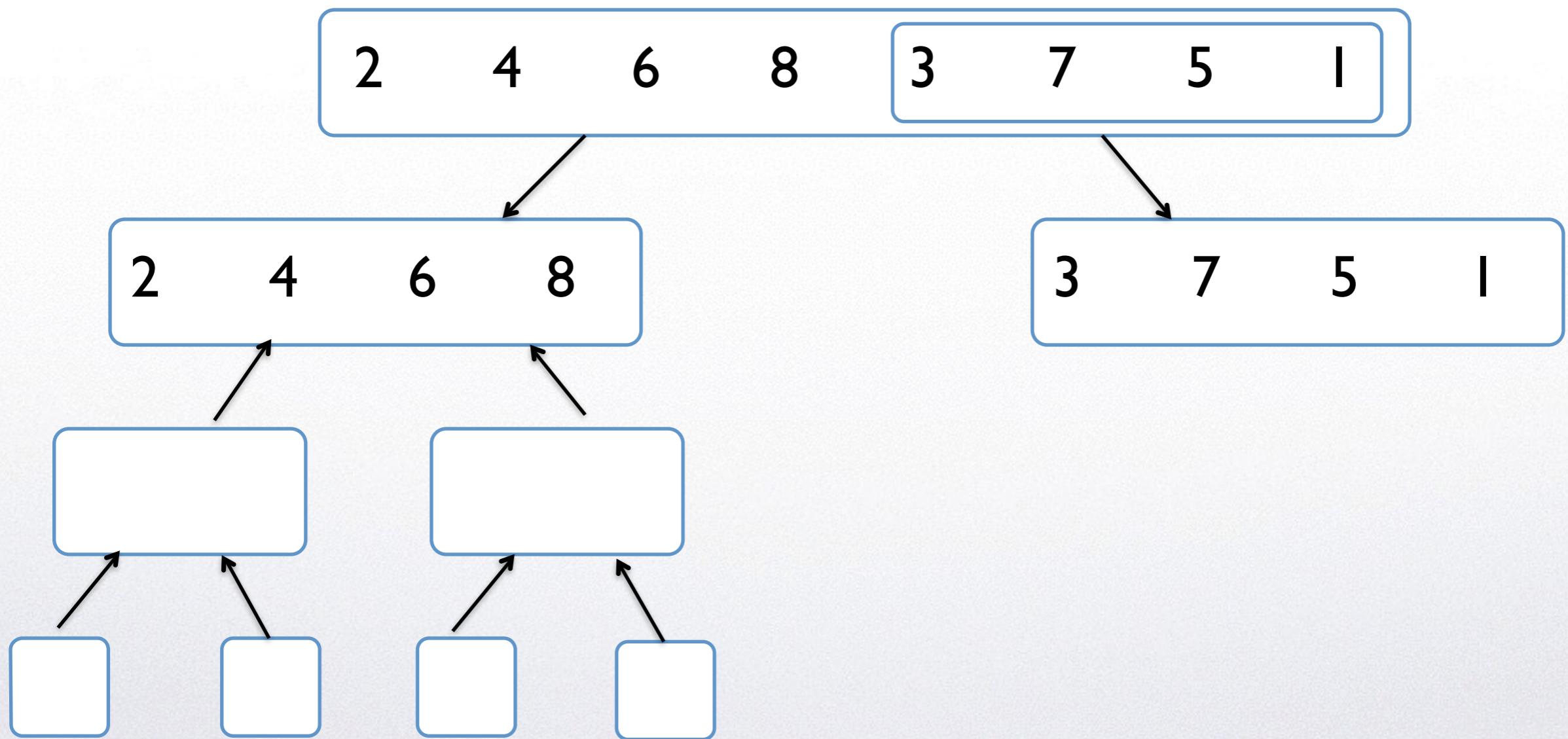


Merge (fase dividir)





Merge (fase dividir)





Revisão?



Perguntas?



"...Embora nós, seres humanos, sejamos muito limitados fisicamente, nossas mentes estão livres para explorar todo o universo..." Stephen Hawking

- O universo em uma casca de nós