

# ORDENAÇÃO: *MERGESORT*

ALGORITMOS E  
ESTRUTURAS DE DADOS II

PUC MINAS

ENGENHARIA DE SOFTWARE

# MERGESORT

- Algoritmo **recursivo**.
- Usa o princípio da **intercalação** (*merge*) para ordenar os dados.
- Princípios básicos:
  - divisão **recursiva** em subvetores;
  - posterior **intercalação** desses subvetores.

# MERGESORT

95	15	42	4	23	16	8	77
----	----	----	---	----	----	---	----

# MERGESORT

95	15	42	4	23	16	8	77
----	----	----	---	----	----	---	----

95	15	42	4
----	----	----	---

23	16	8	77
----	----	---	----

# MERGESORT

95	15	42	4	23	16	8	77
----	----	----	---	----	----	---	----

95	15	42	4
----	----	----	---

23	16	8	77
----	----	---	----

95	15
----	----

42	4
----	---

23	16
----	----

8	77
---	----

# MERGESORT

95	15	42	4	23	16	8	77
----	----	----	---	----	----	---	----

95	15	42	4	23	16	8	77
----	----	----	---	----	----	---	----

95	15	42	4	23	16	8	77
----	----	----	---	----	----	---	----

95	15	42	4	23	16	8	77
----	----	----	---	----	----	---	----

# MERGESORT

95

15

42

4

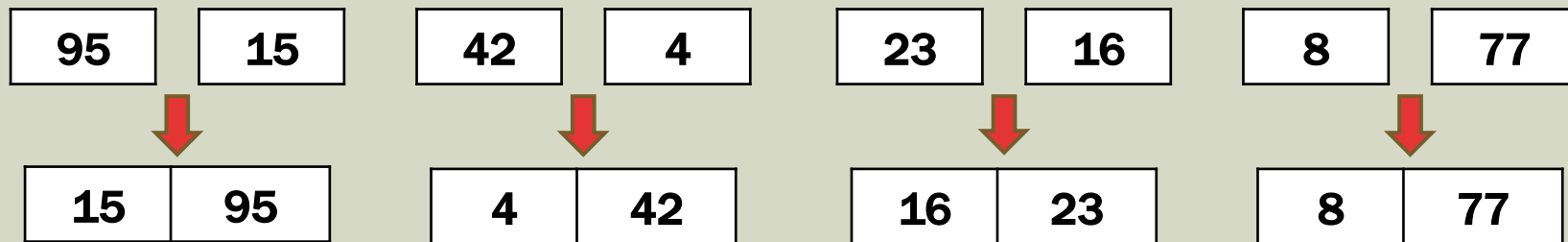
23

16

8

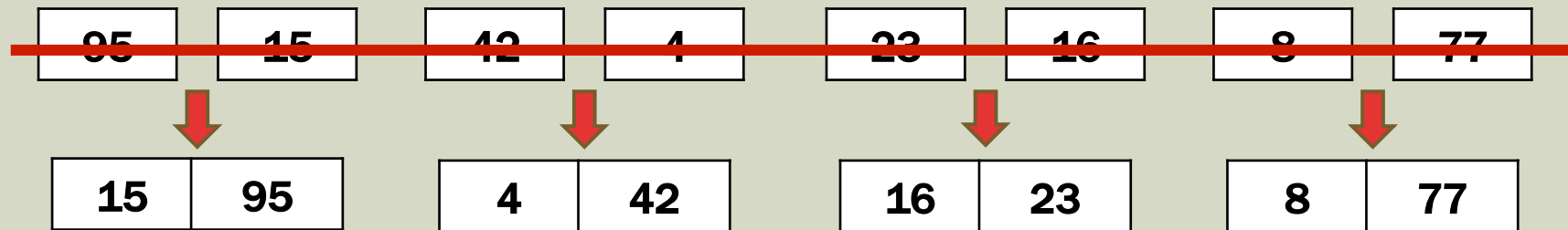
77

# MERGESORT

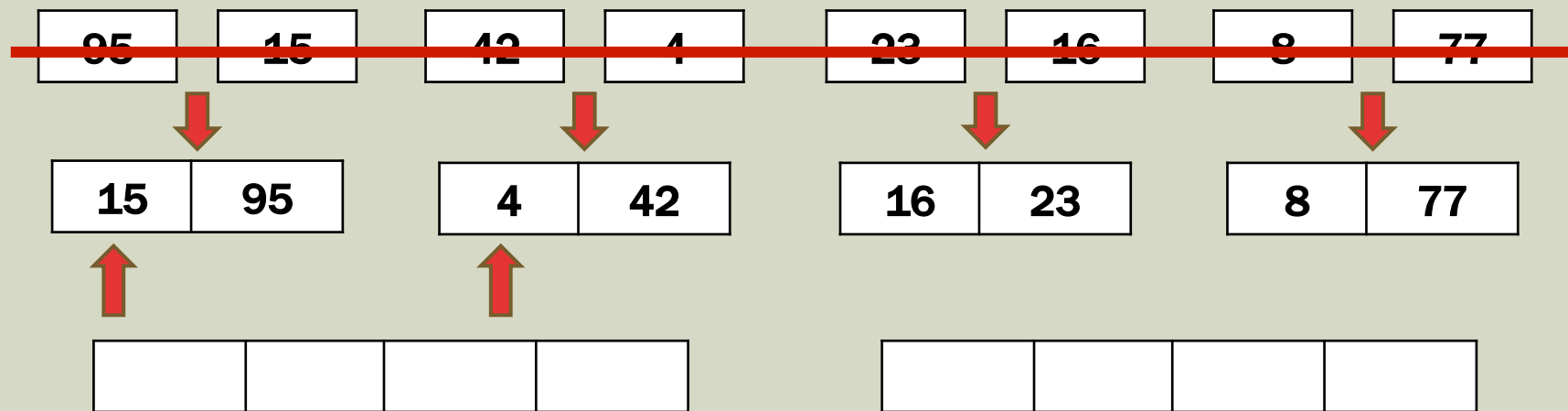




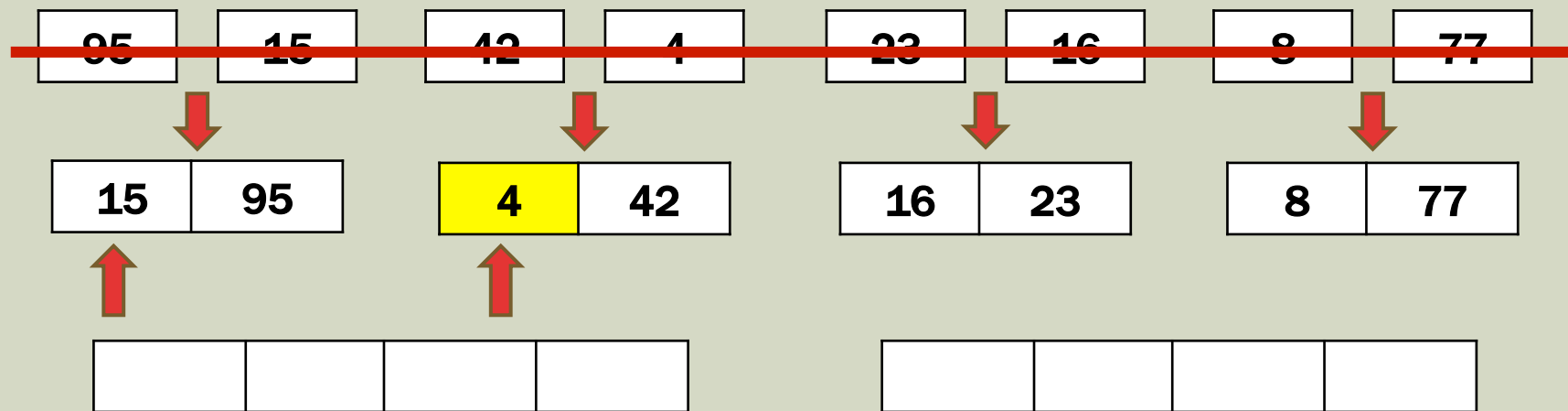
# MERGESORT



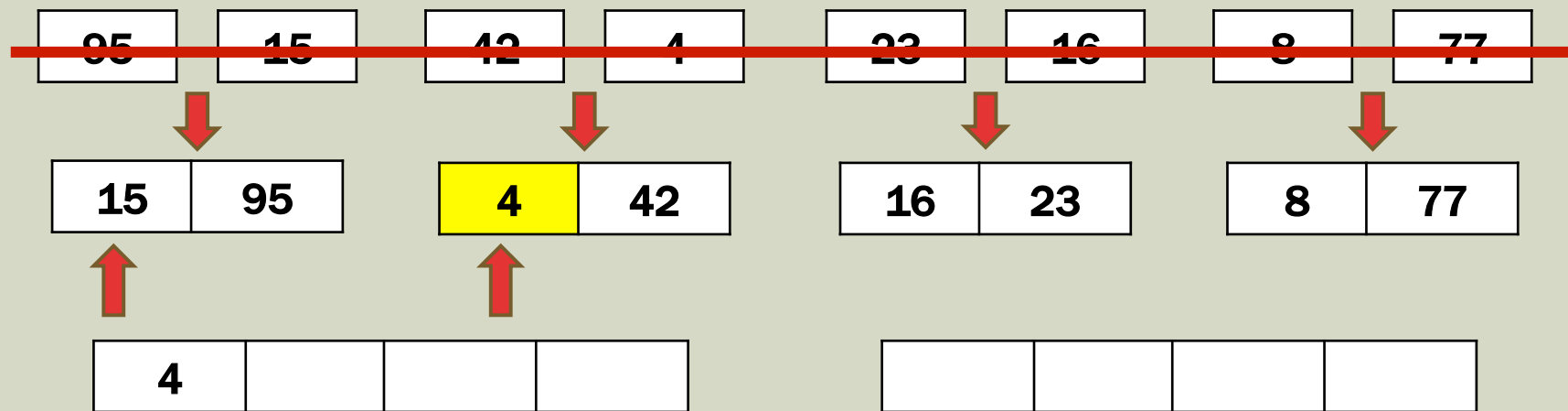
# MERGESORT



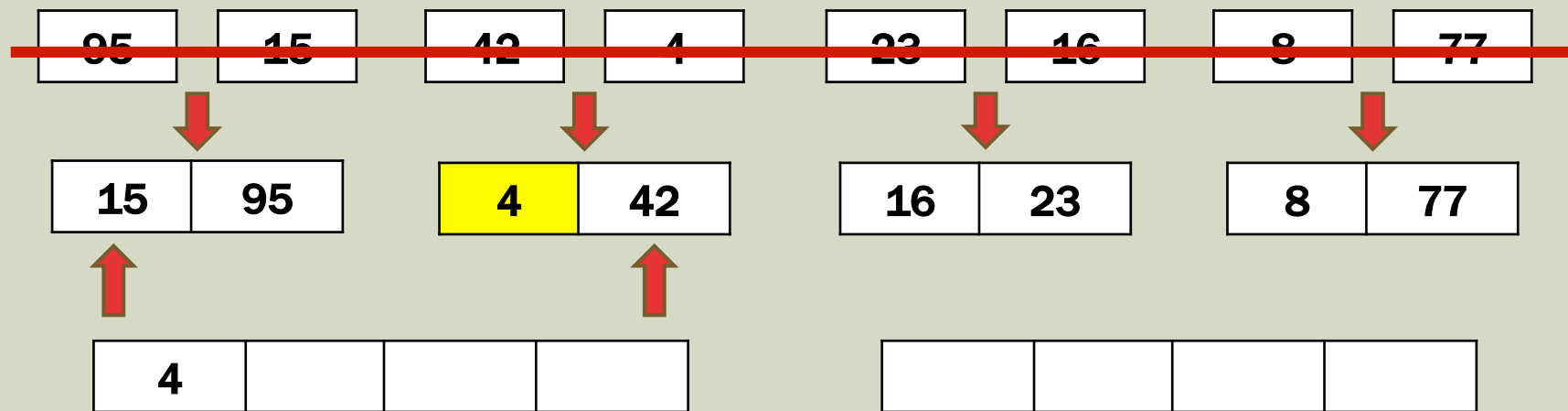
# MERGESORT



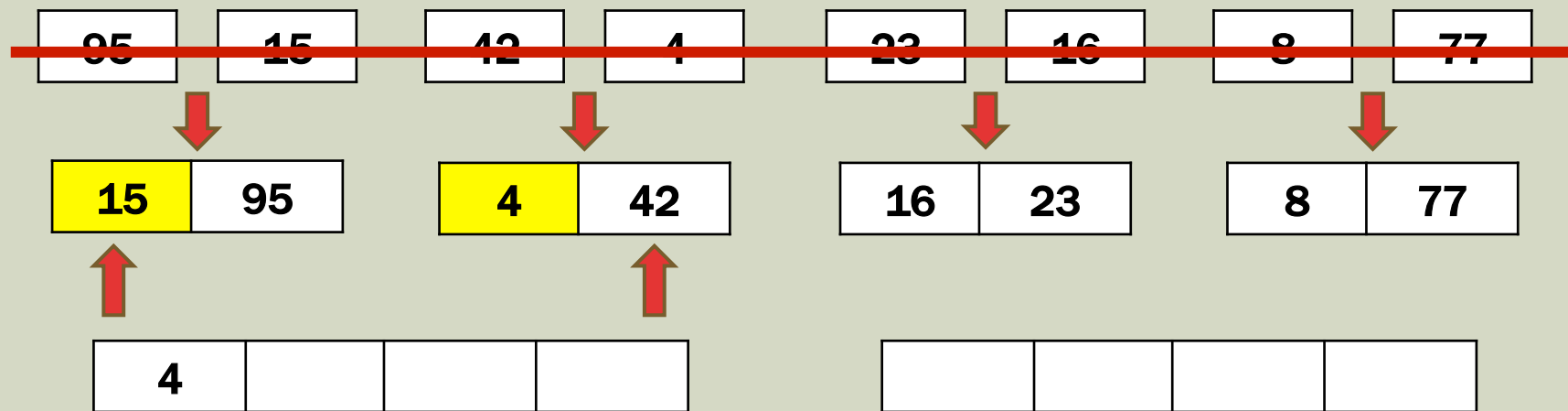
# MERGESORT



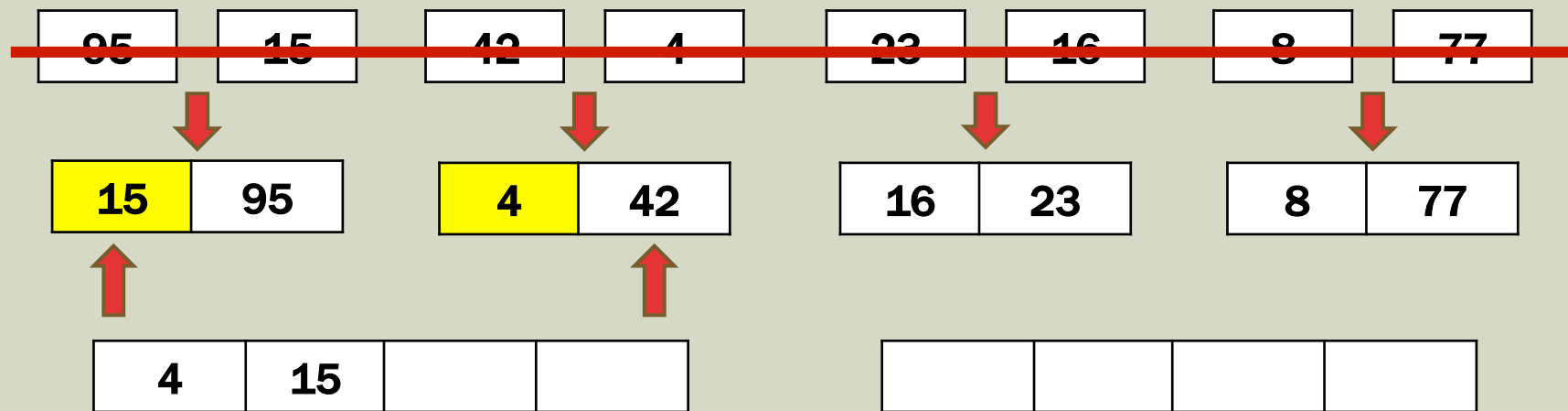
# MERGESORT



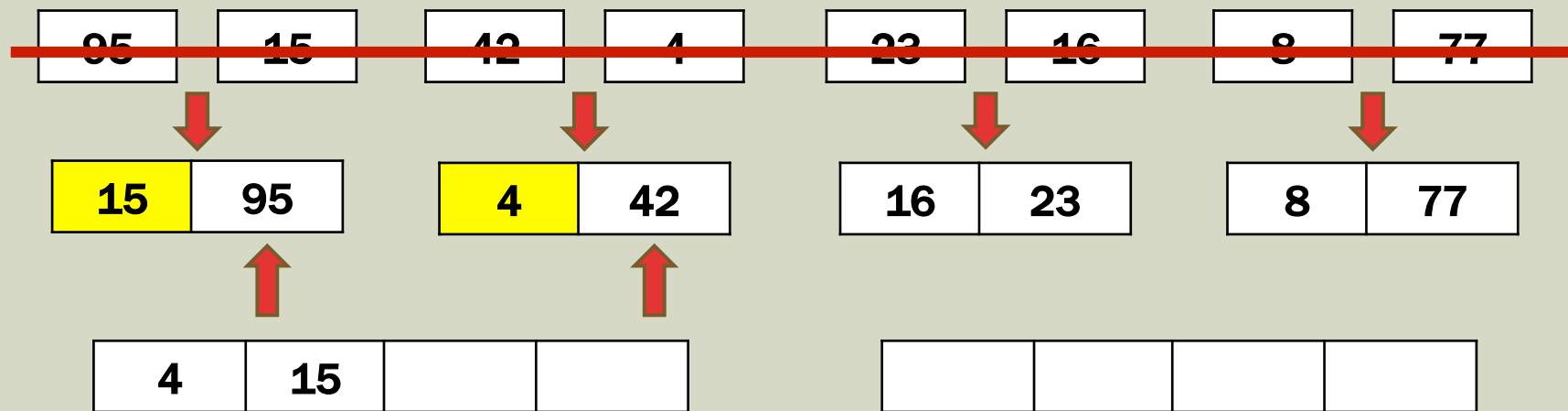
# MERGESORT



# MERGESORT

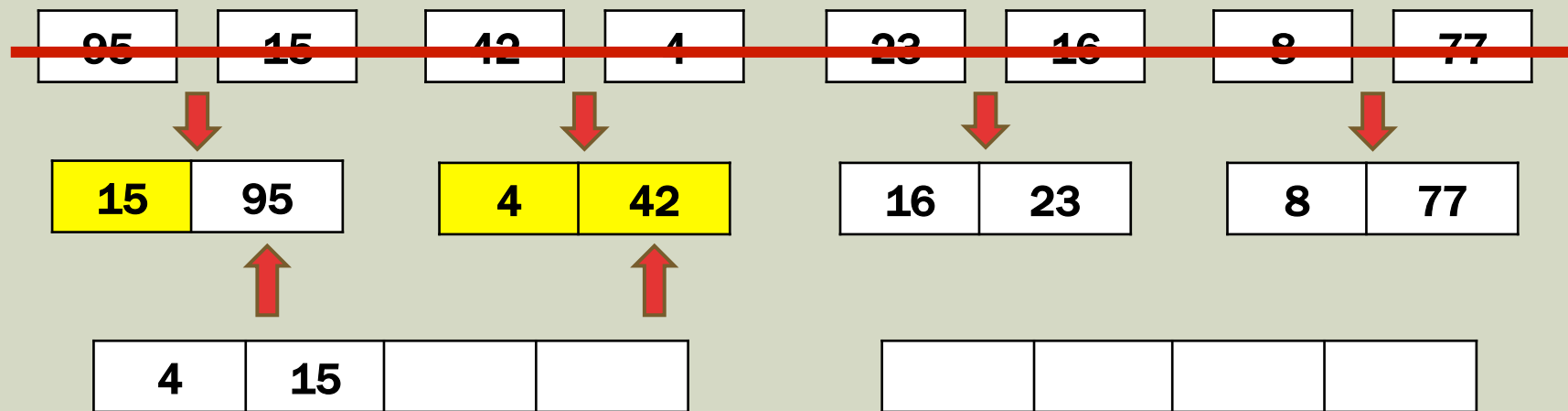


# MERGESORT

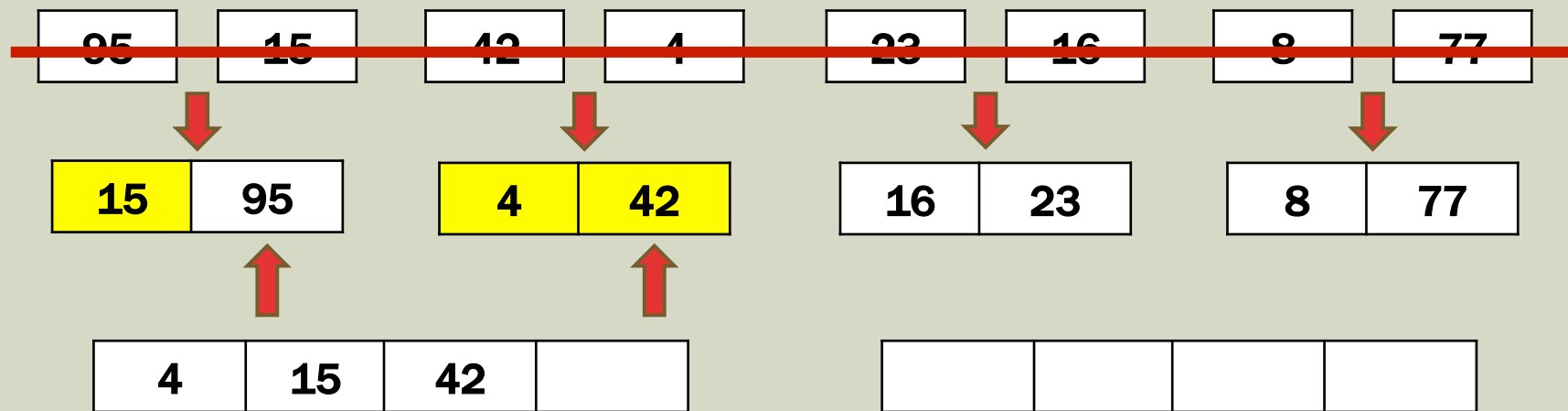




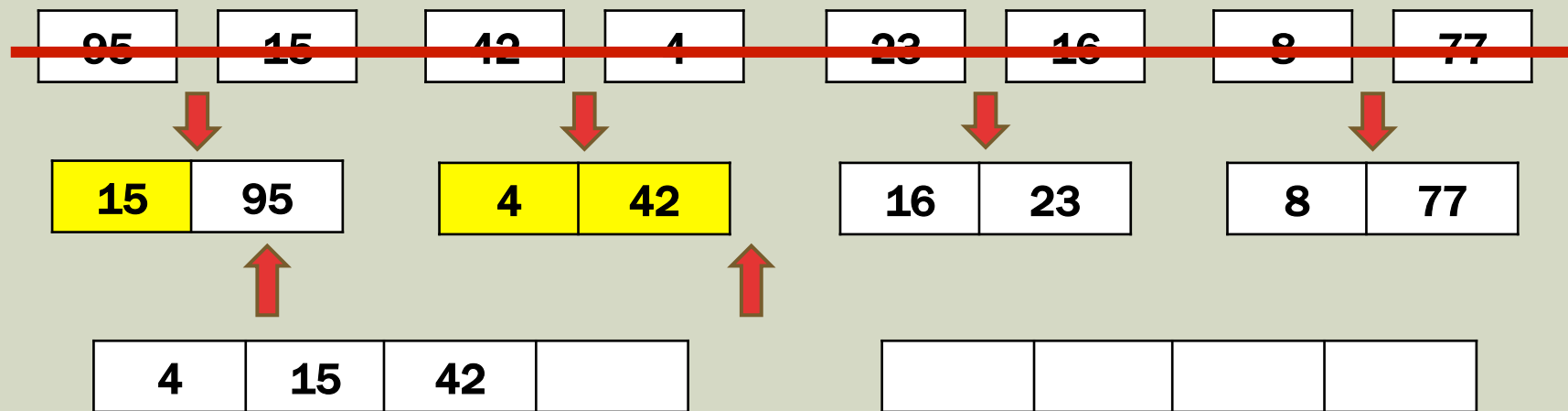
# MERGESORT



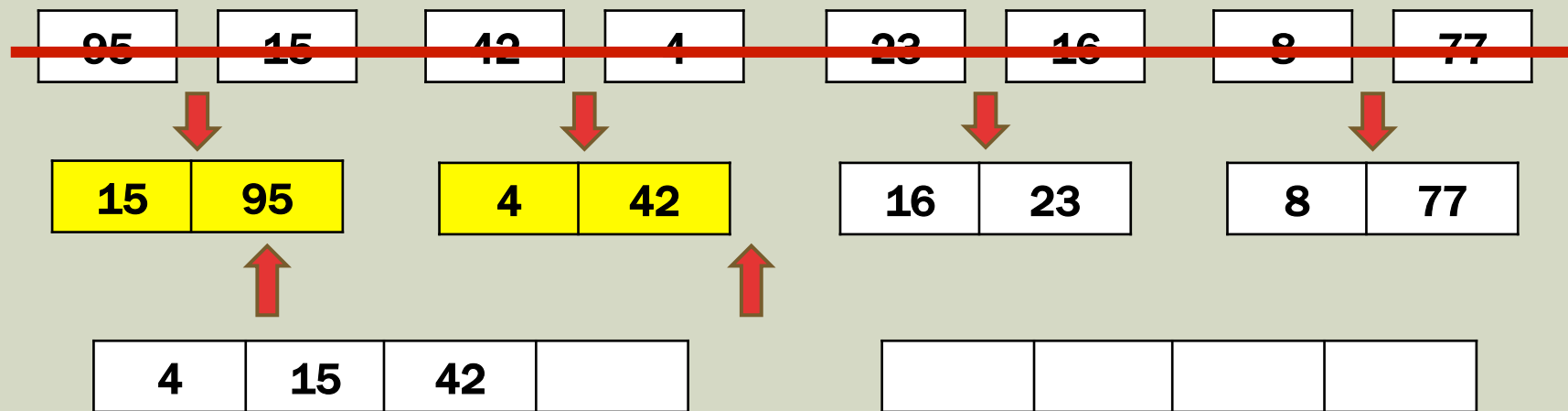
# MERGESORT



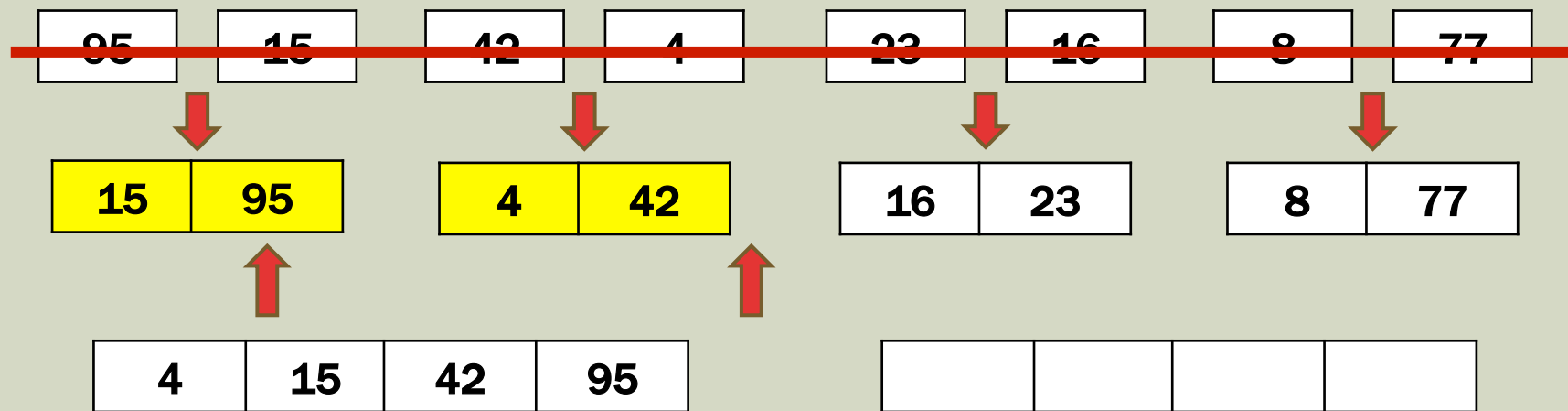
# MERGESORT



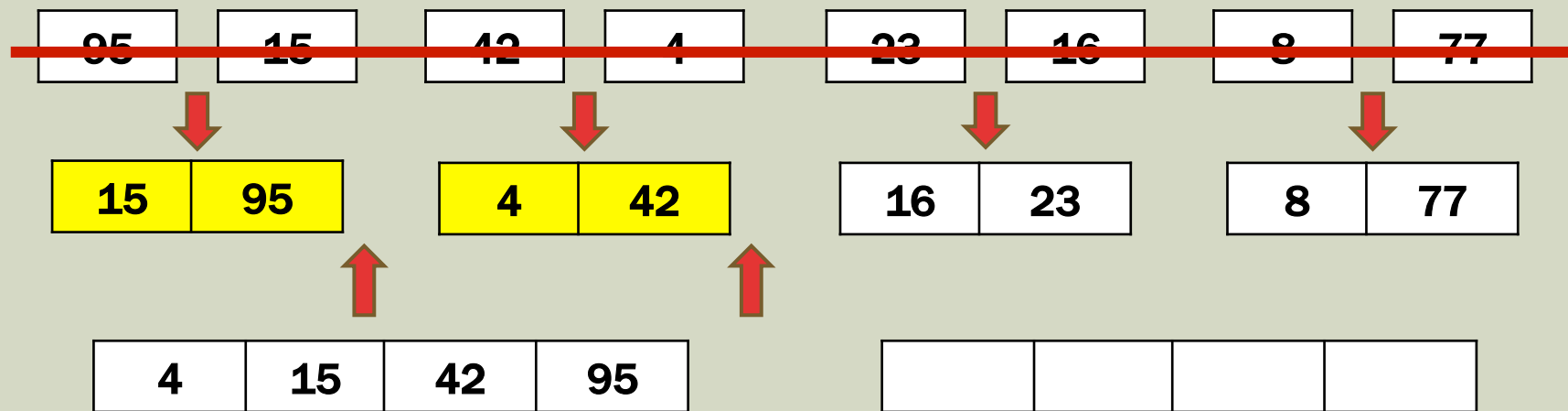
# MERGESORT



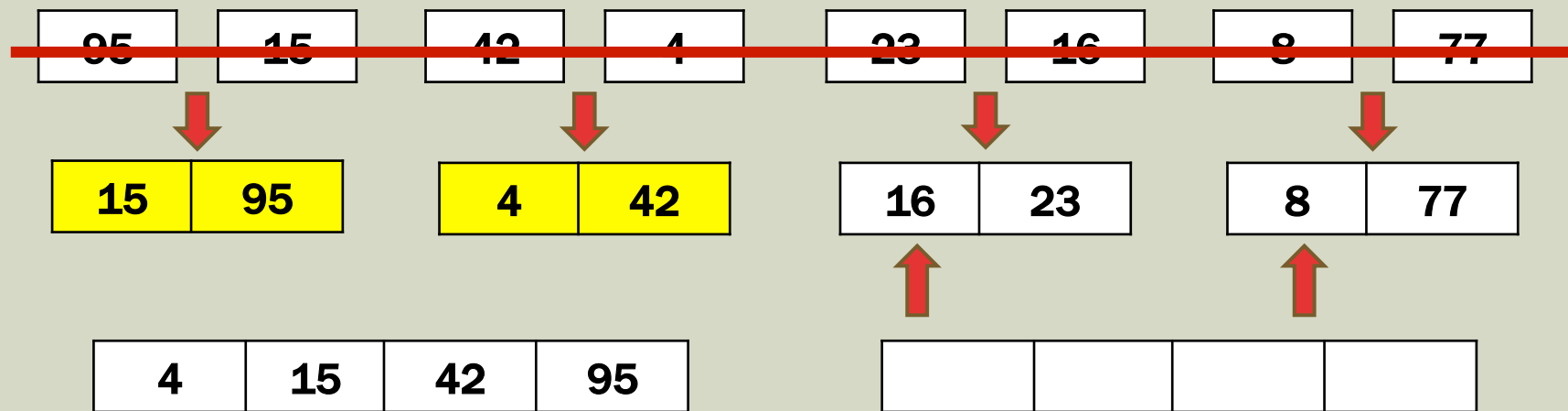
# MERGESORT



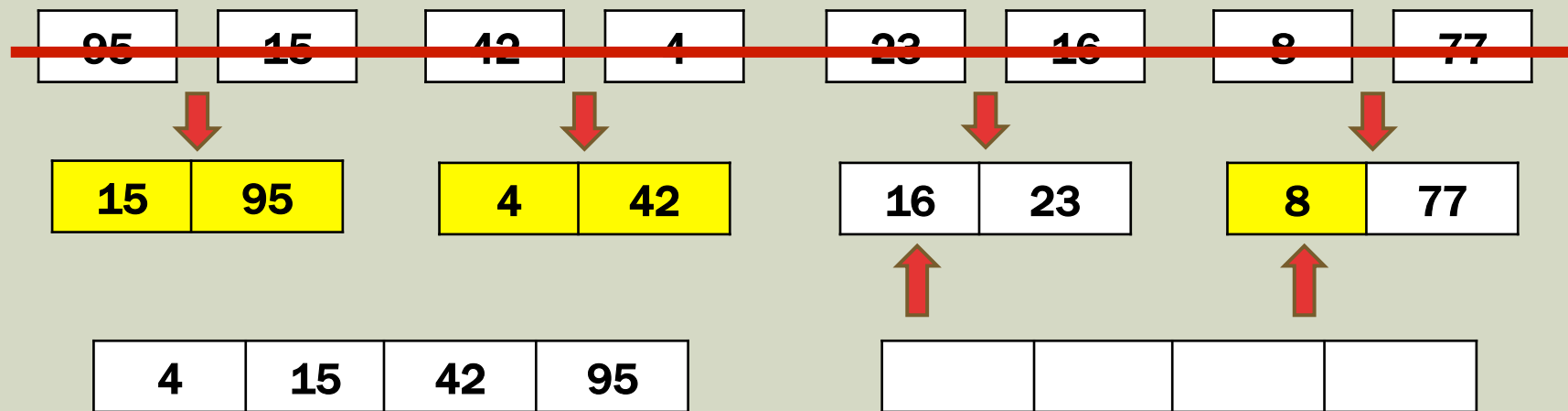
# MERGESORT



# MERGESORT

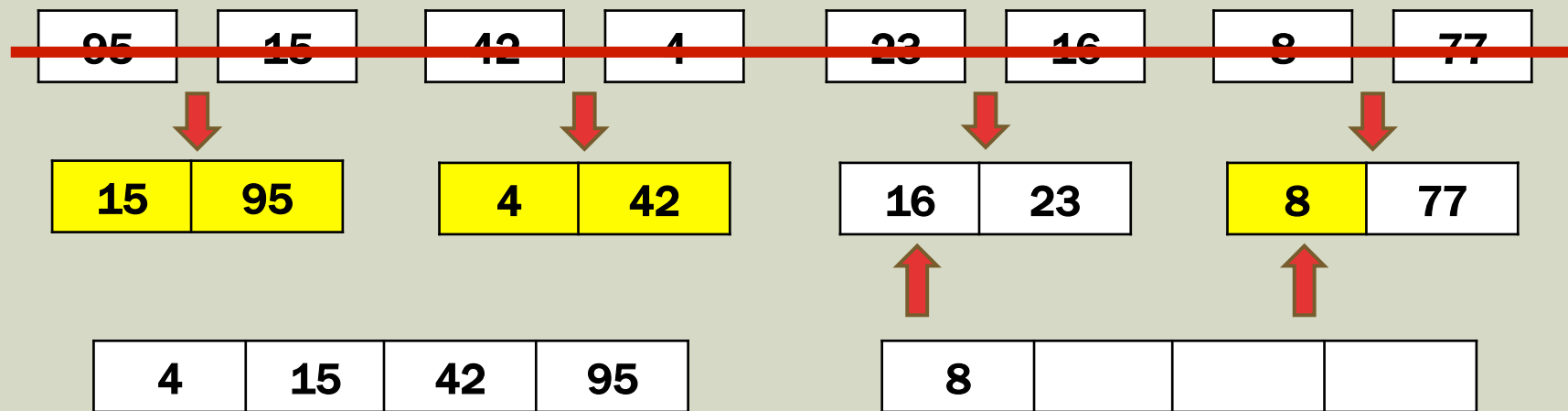


# MERGESORT

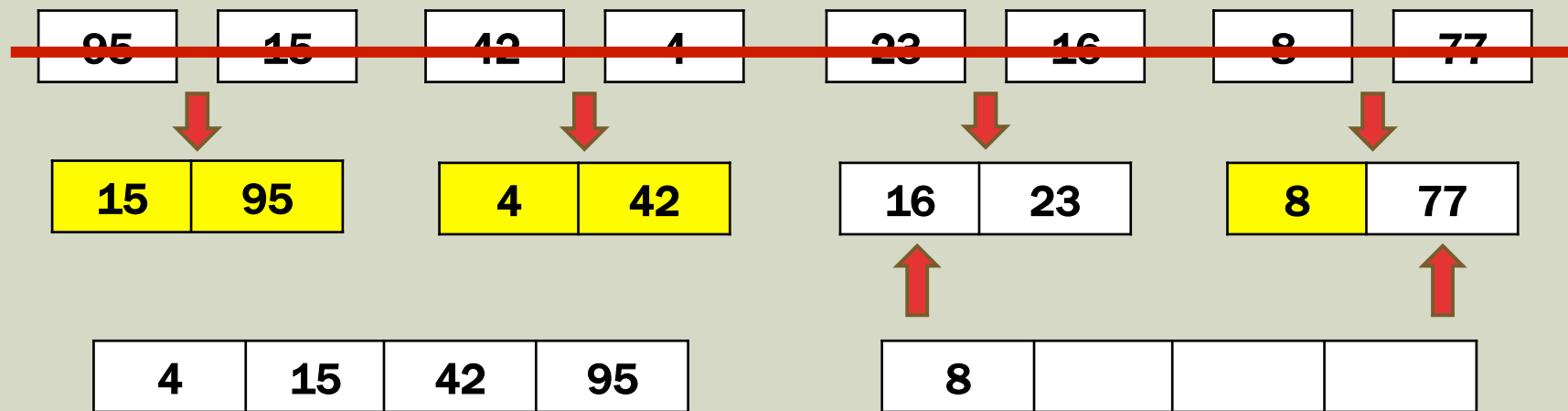




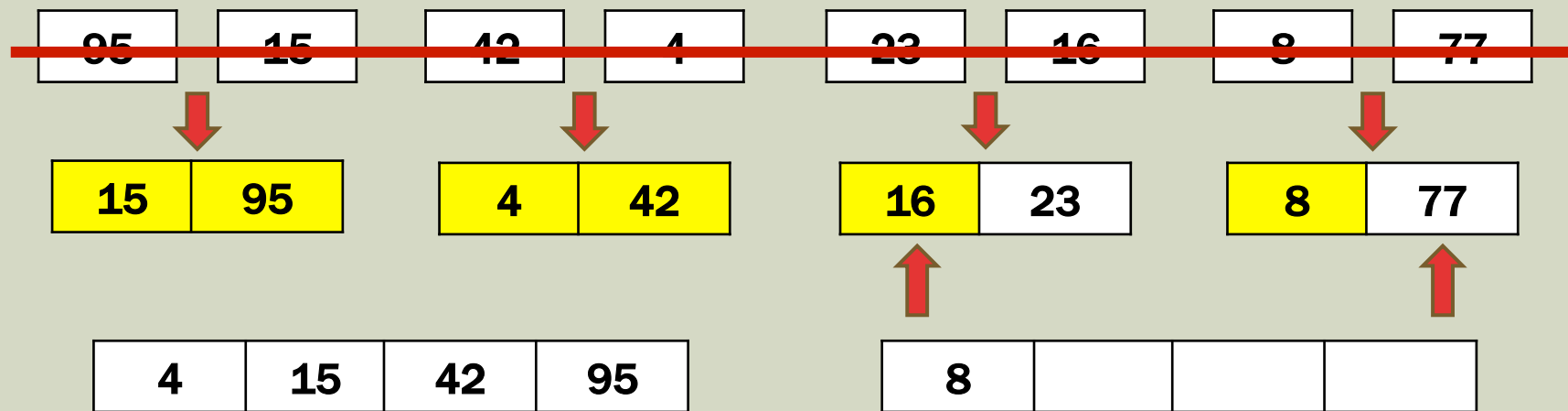
# MERGESORT



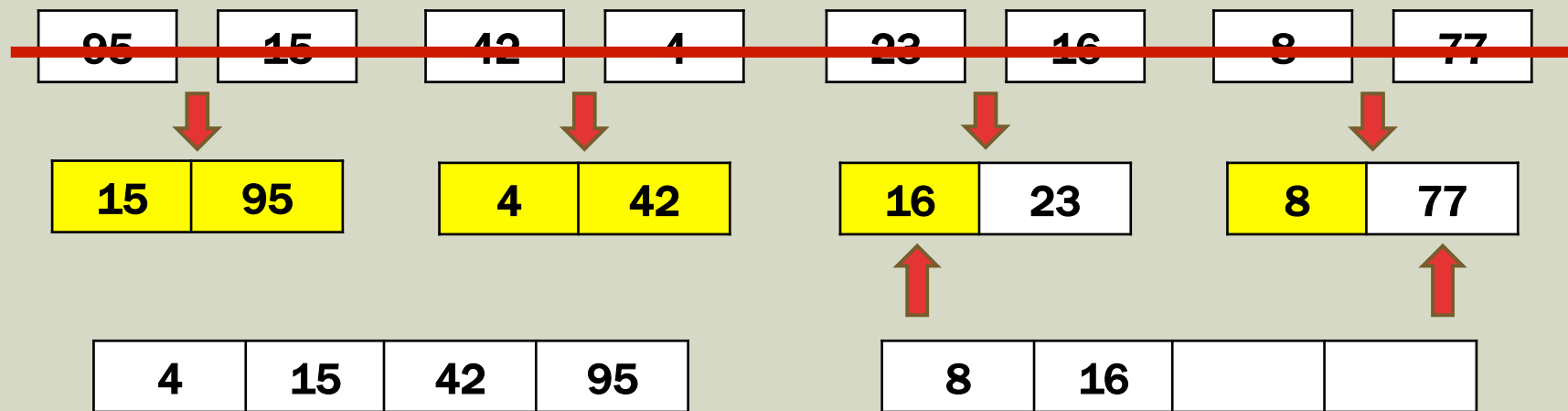
# MERGESORT



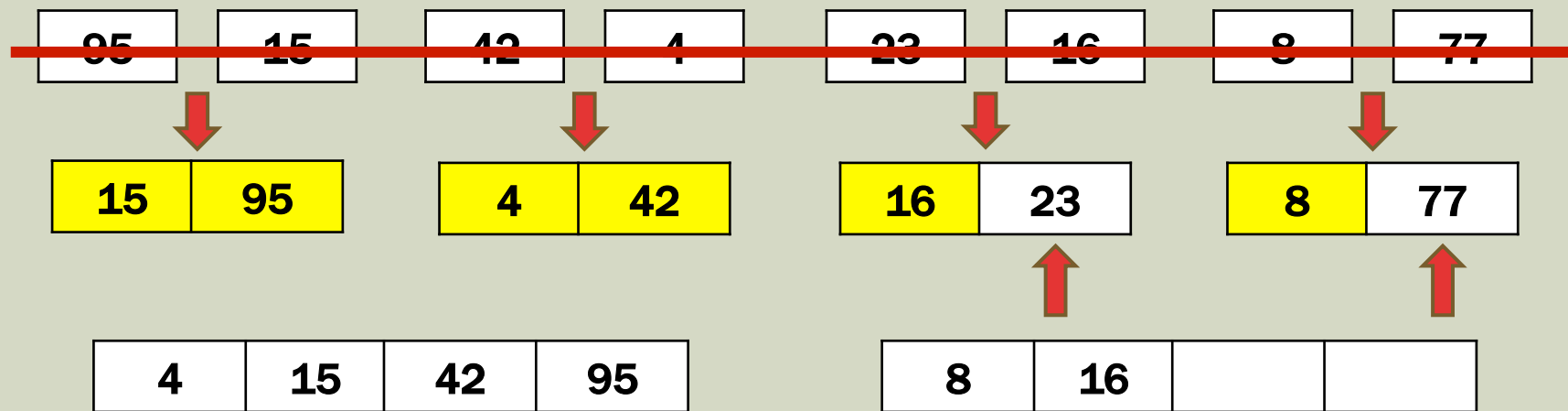
# MERGESORT



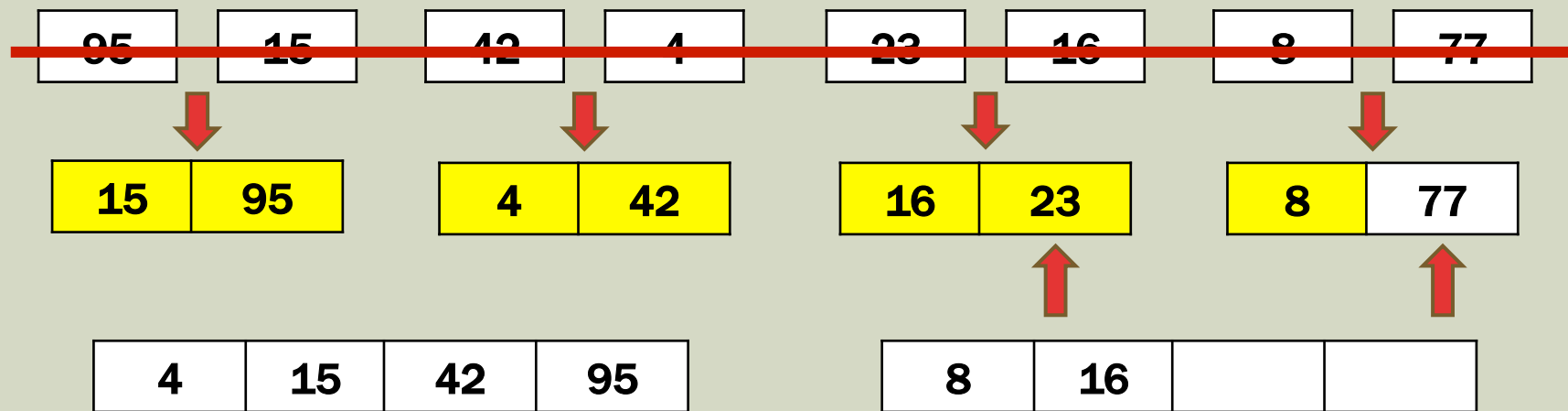
# MERGESORT



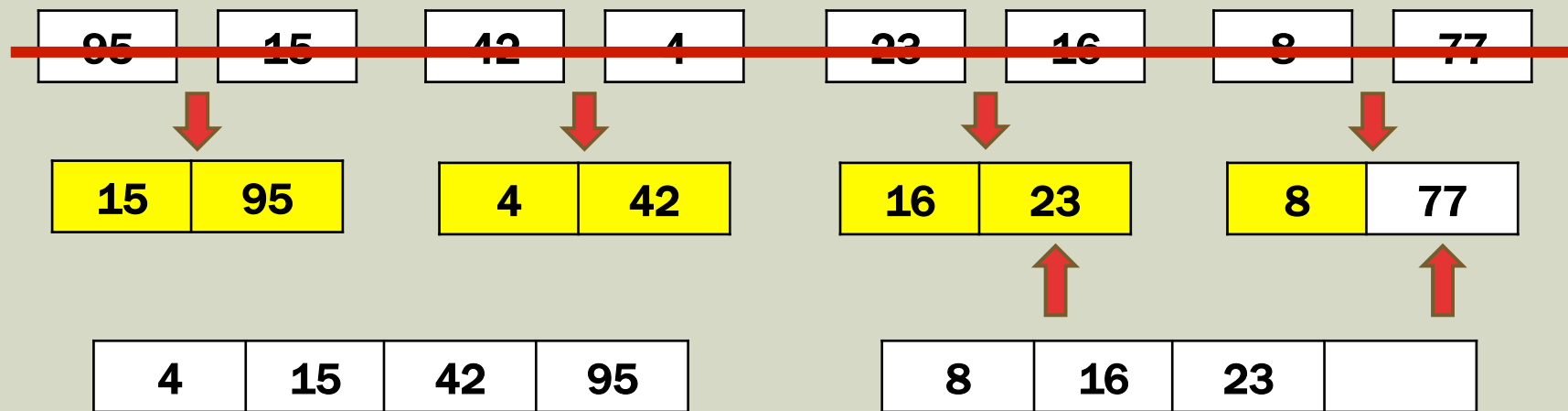
# MERGESORT



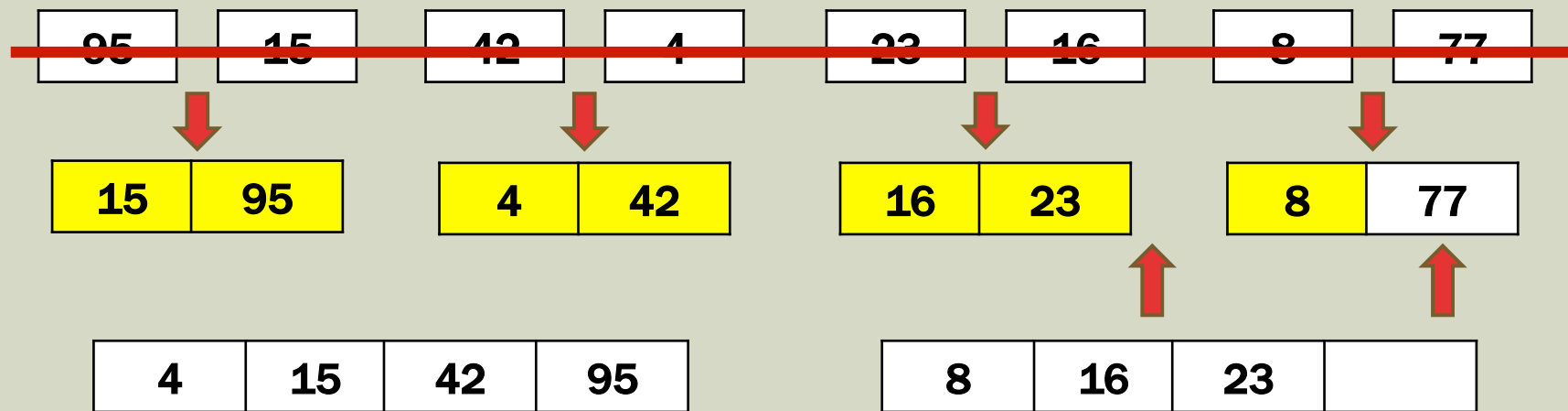
# MERGESORT



# MERGESORT

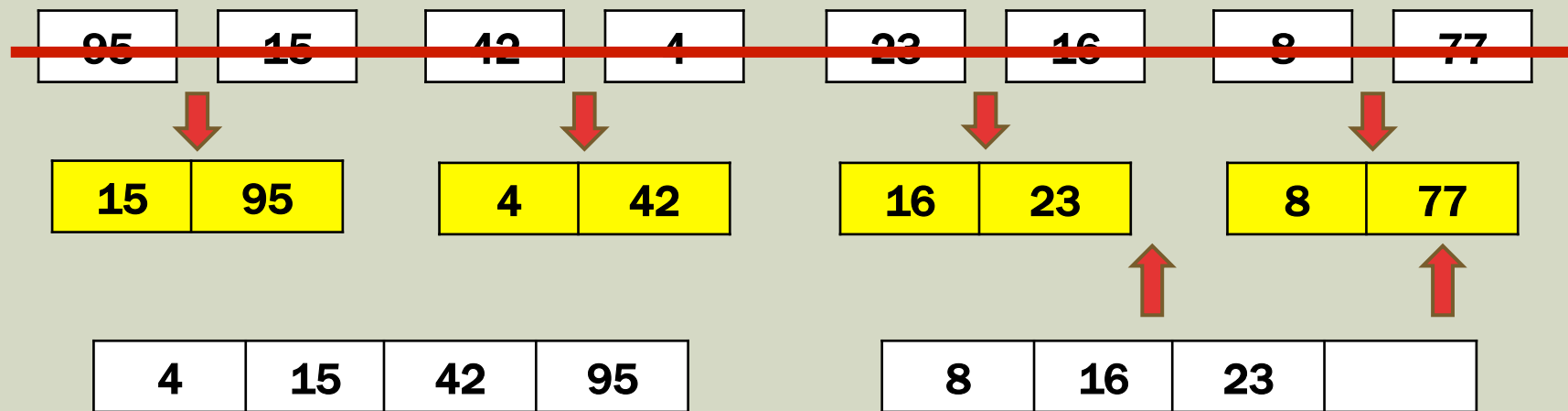


# MERGESORT

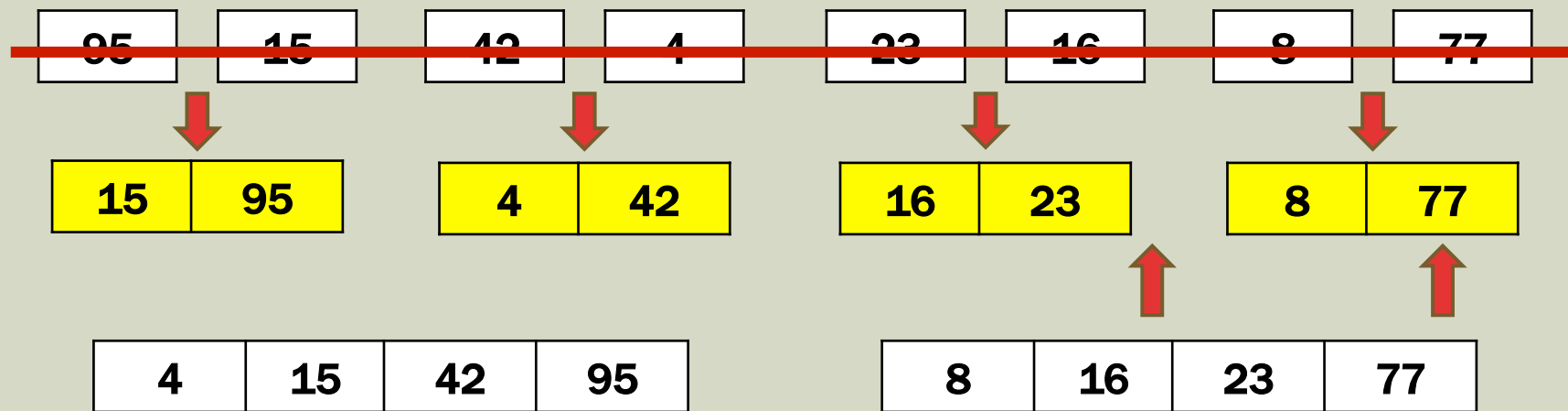




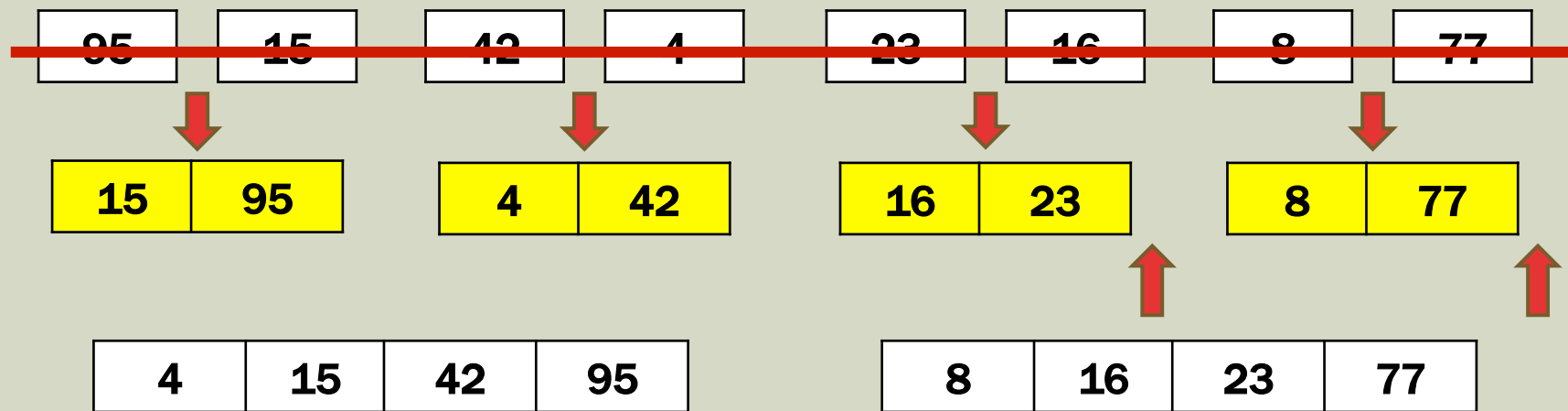
# MERGESORT



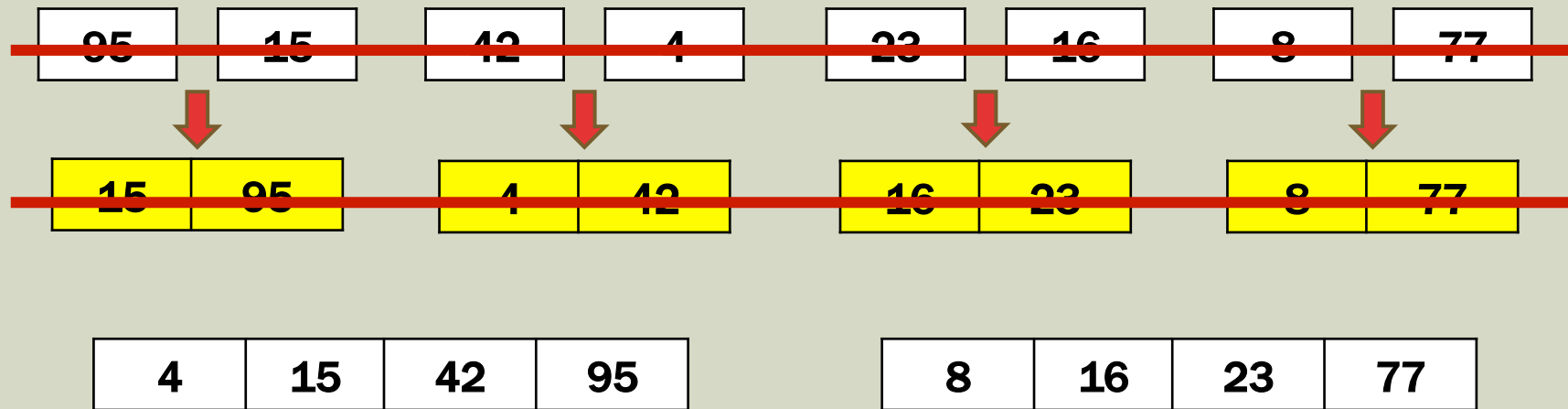
# MERGESORT



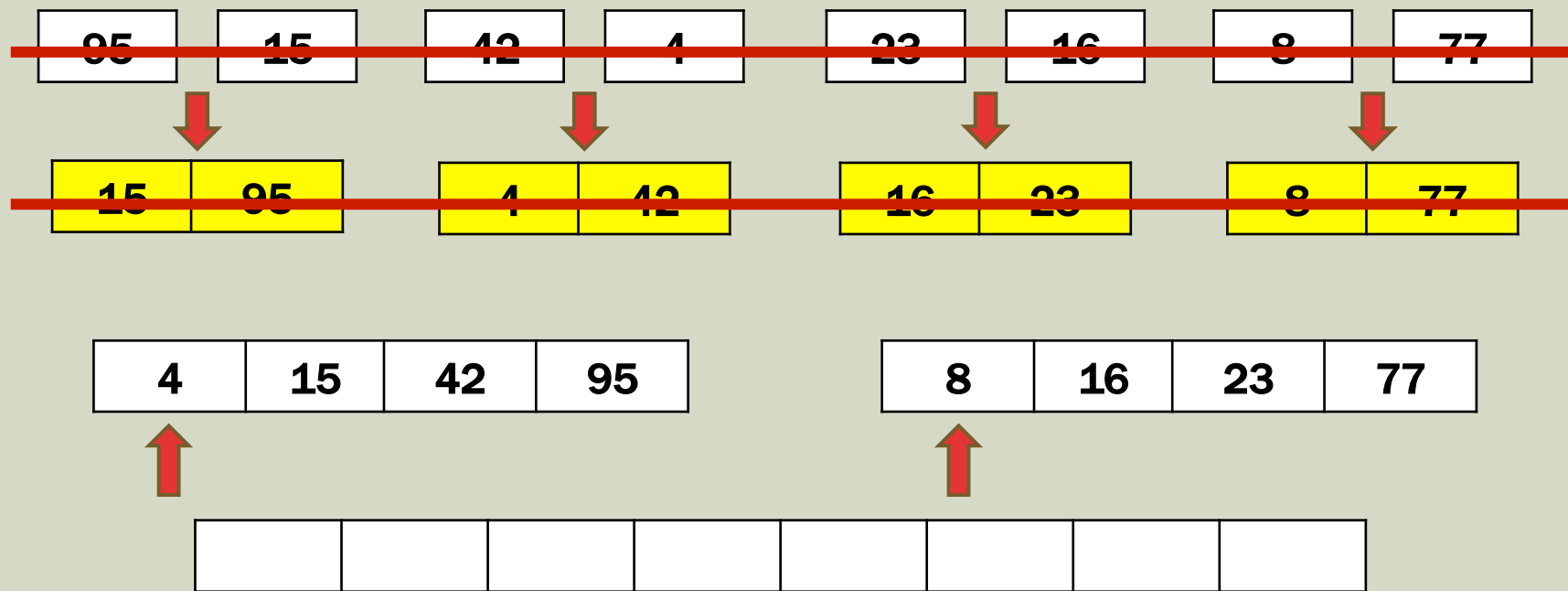
# MERGESORT



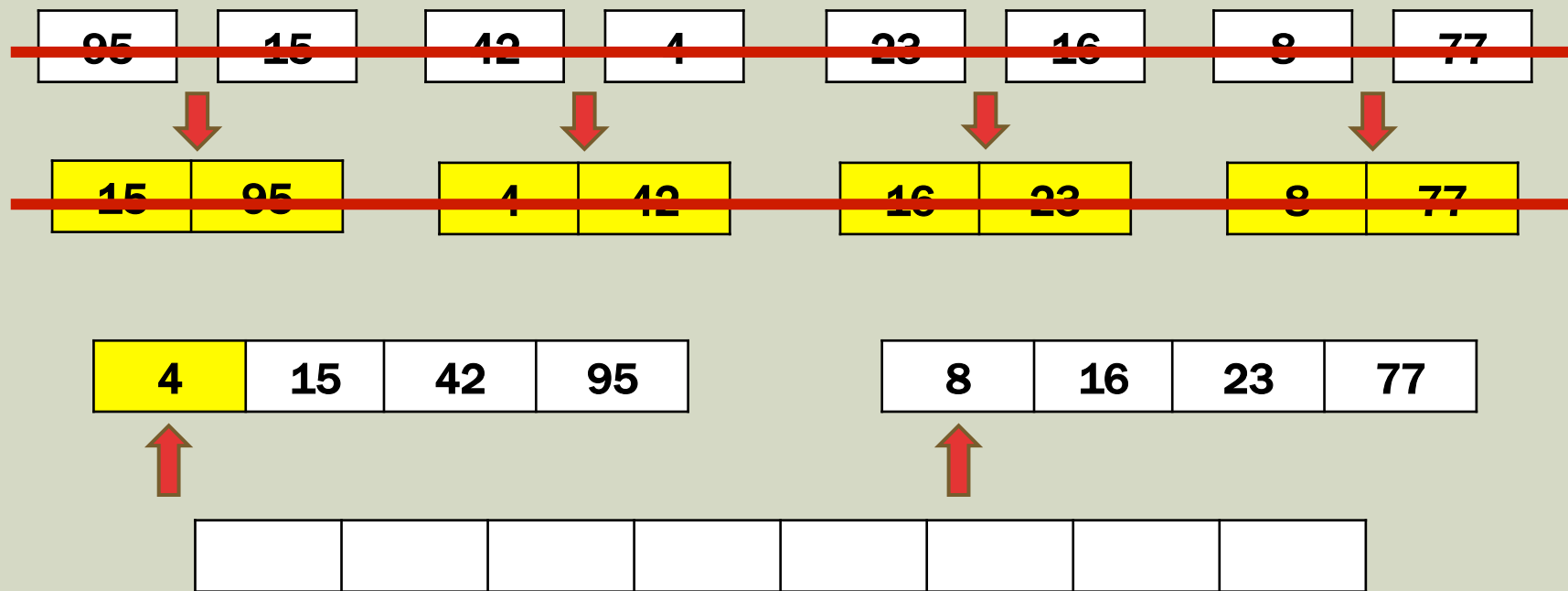
# MERGESORT



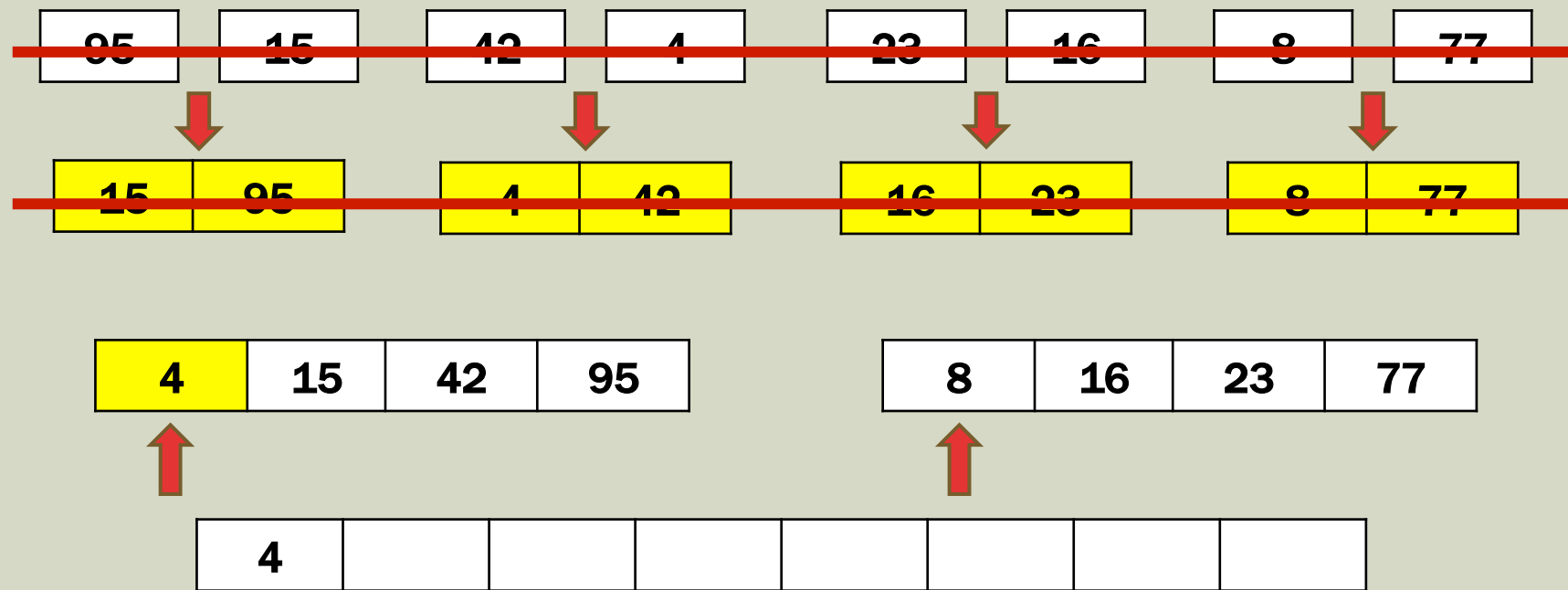
# MERGESORT



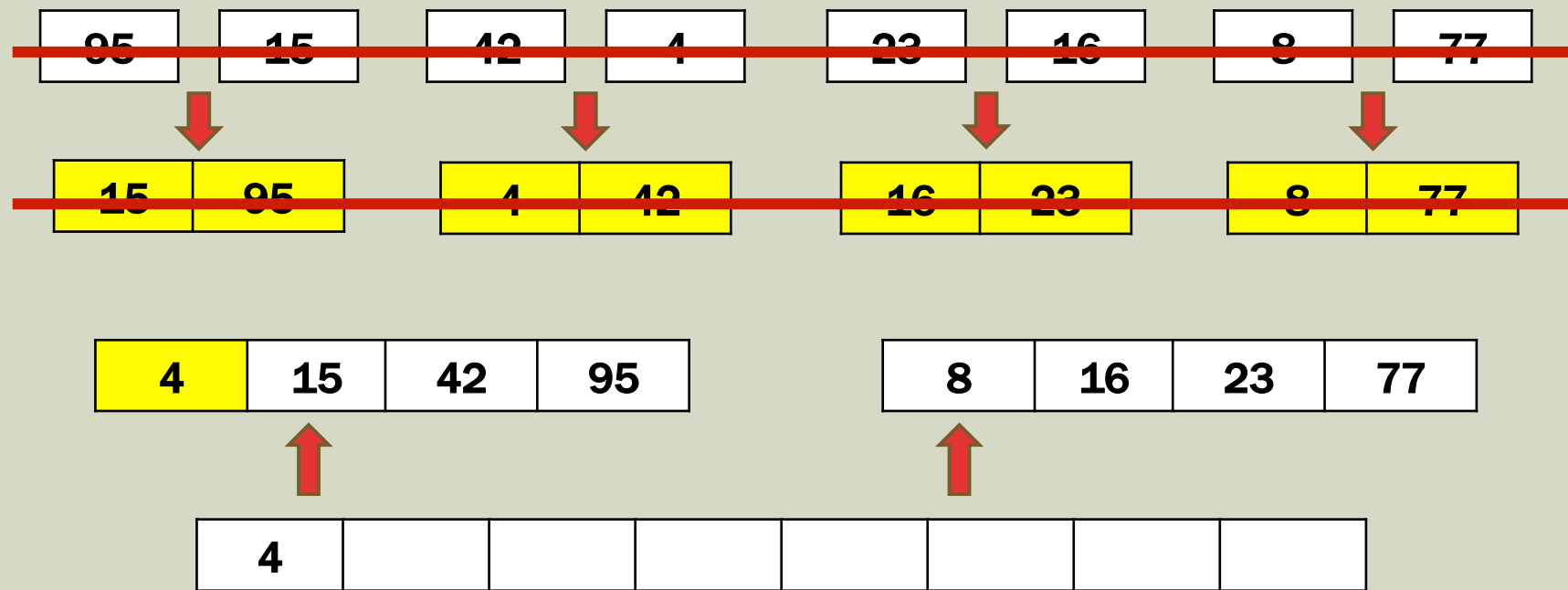
# MERGESORT



# MERGESORT

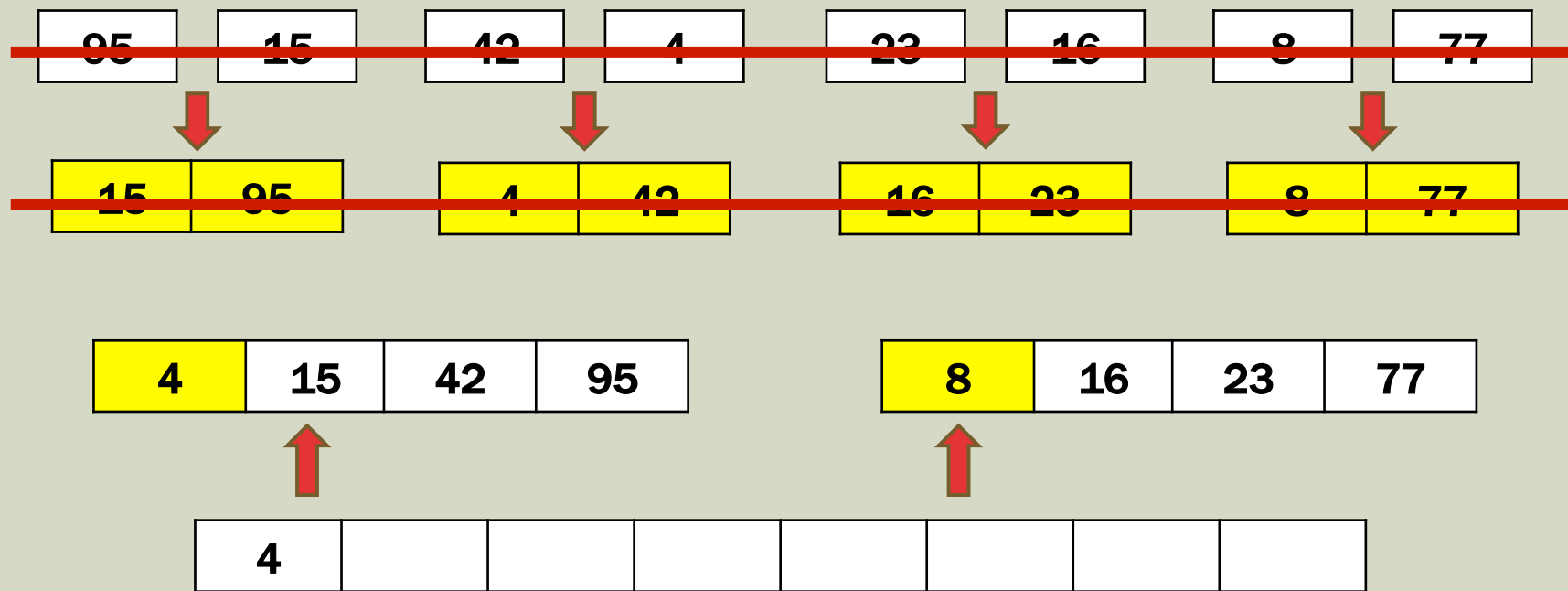


# MERGESORT

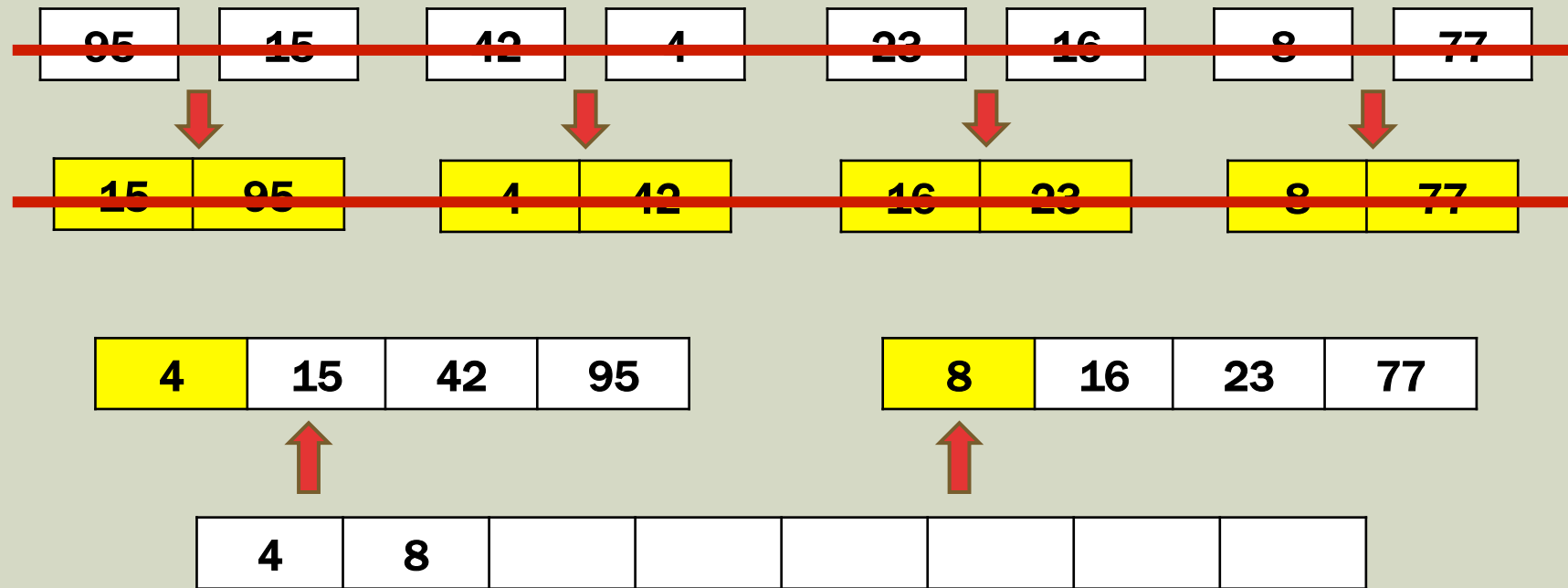




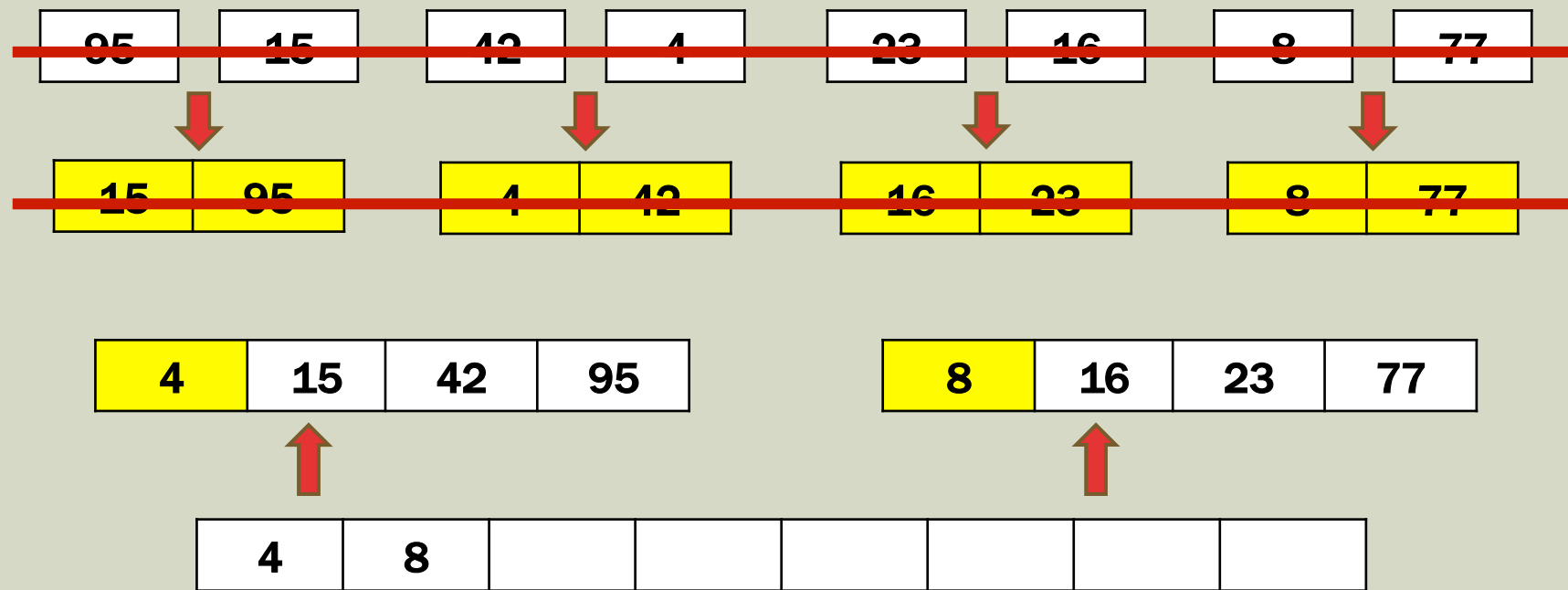
# MERGESORT



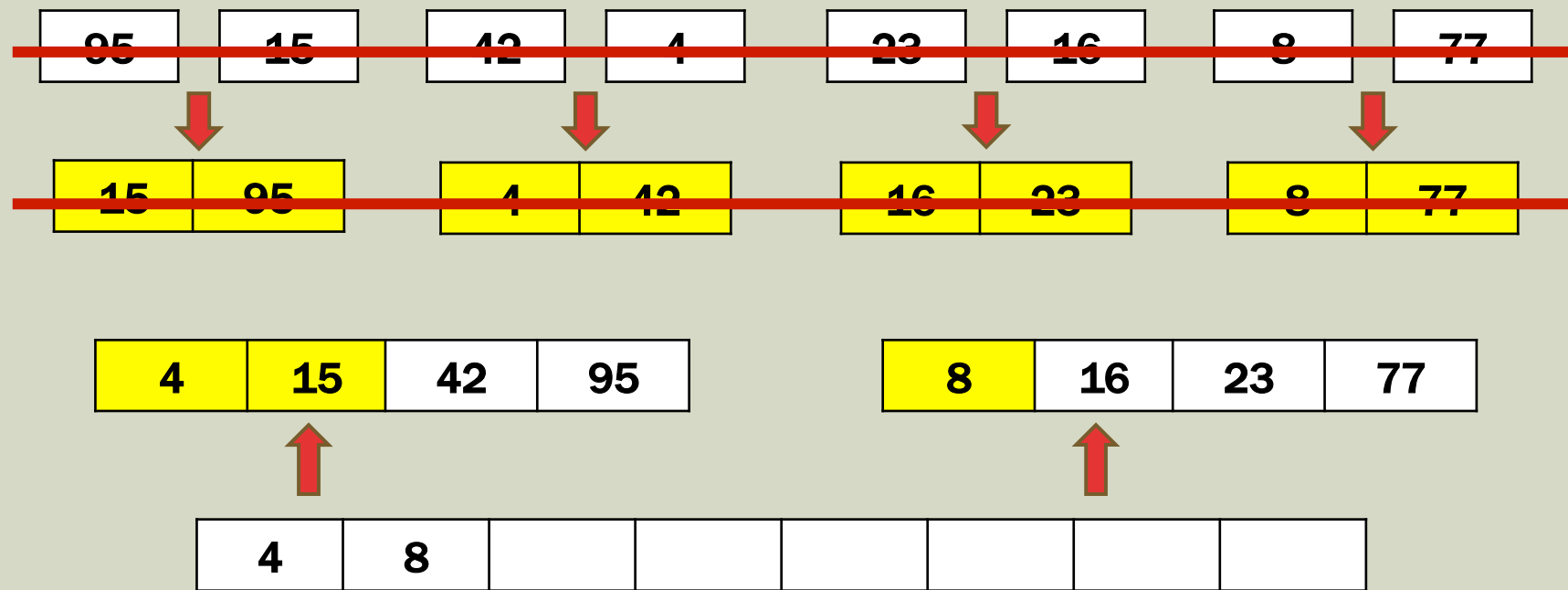
# MERGESORT



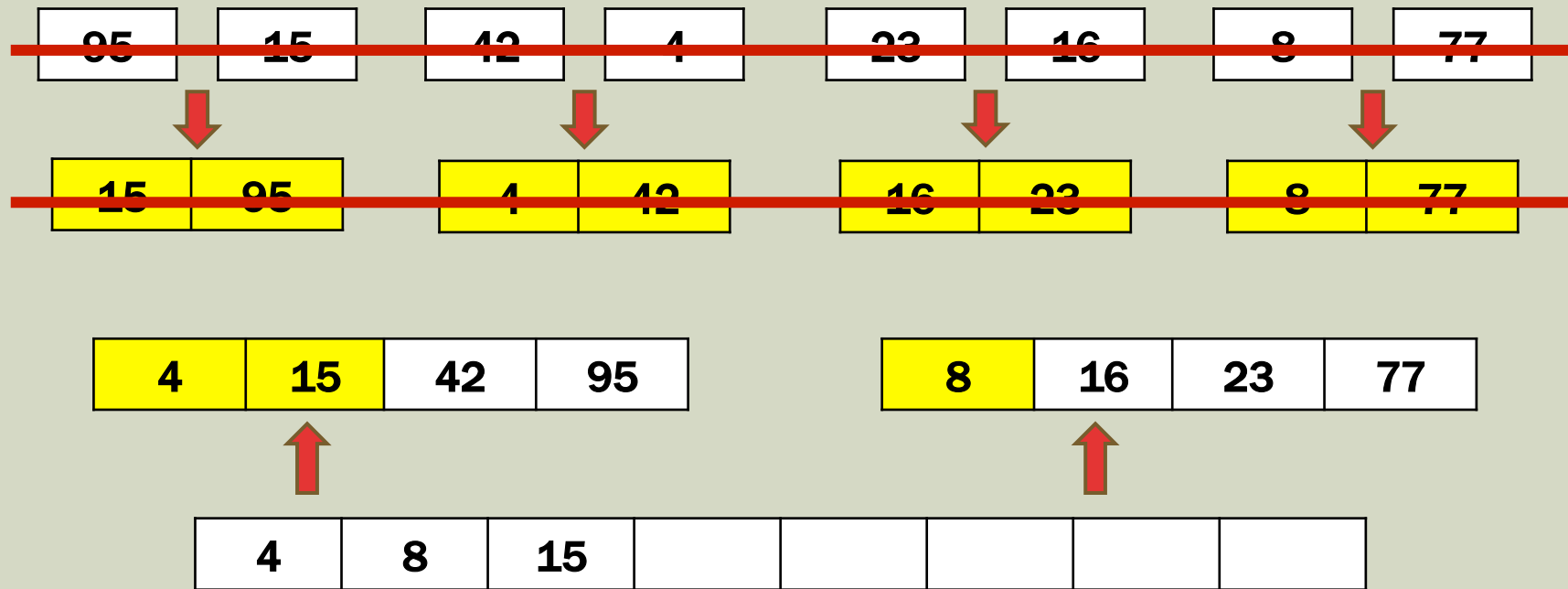
# MERGESORT



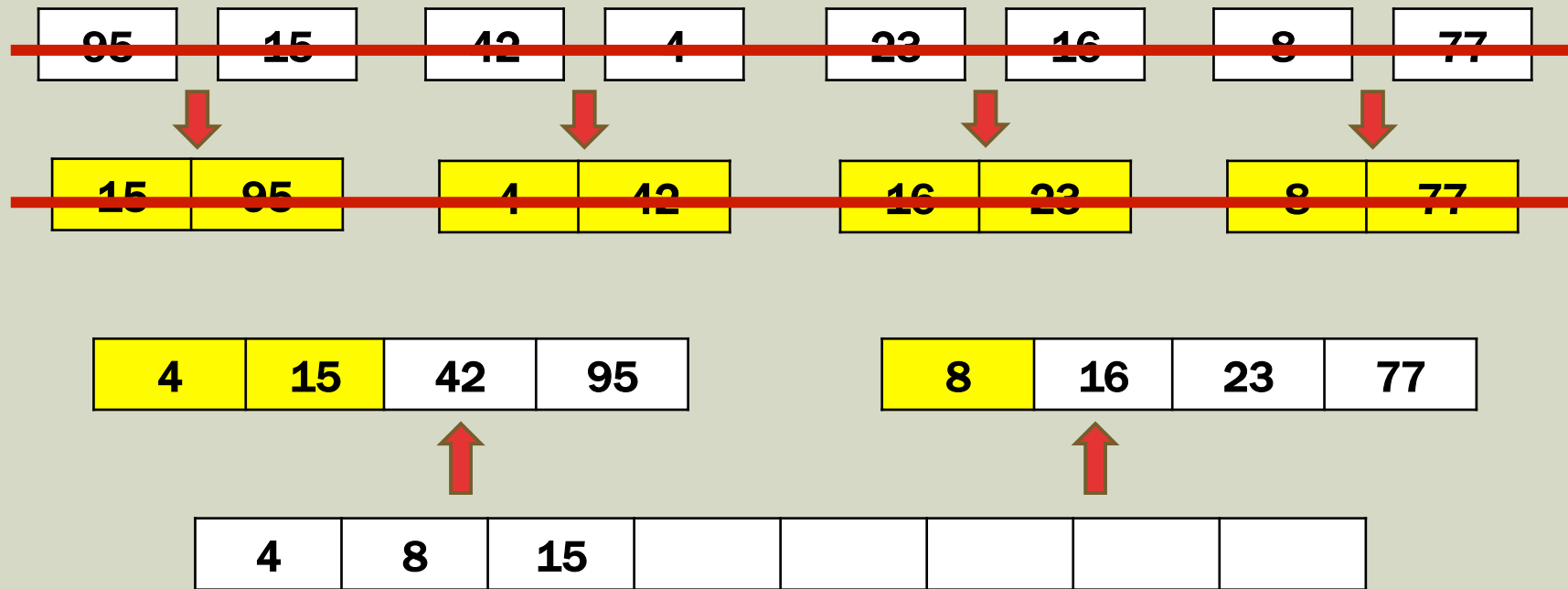
# MERGESORT



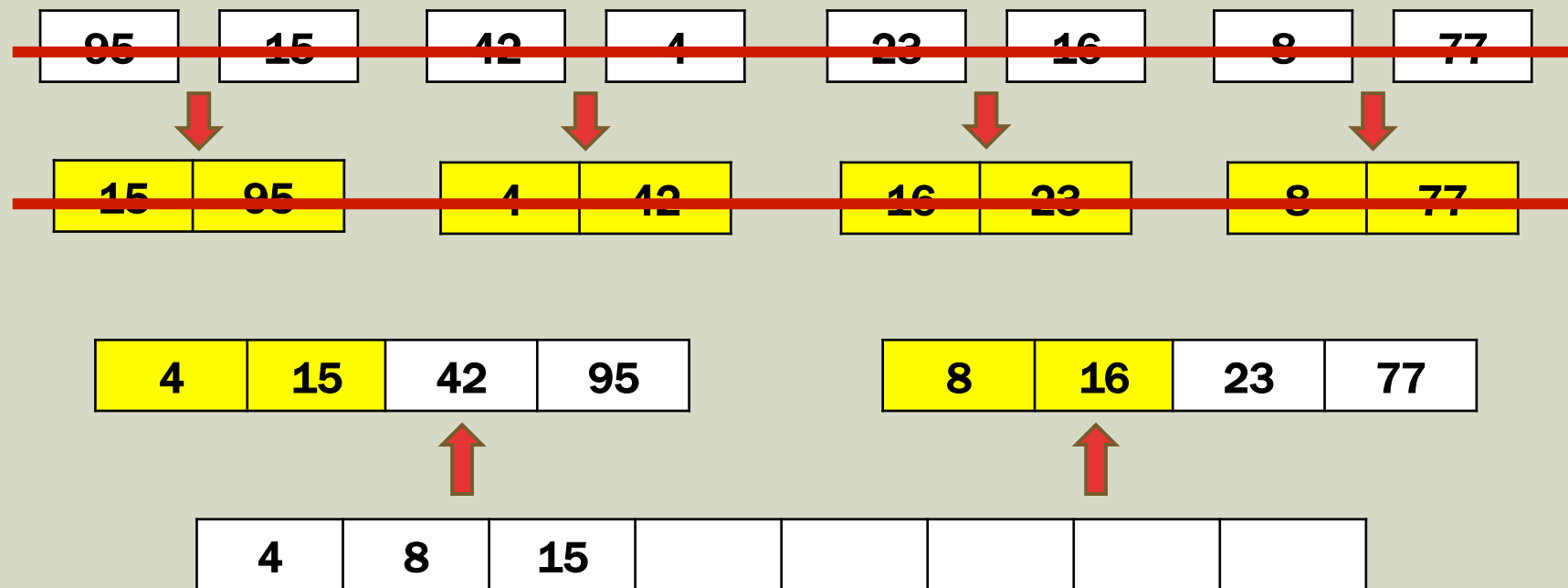
# MERGESORT



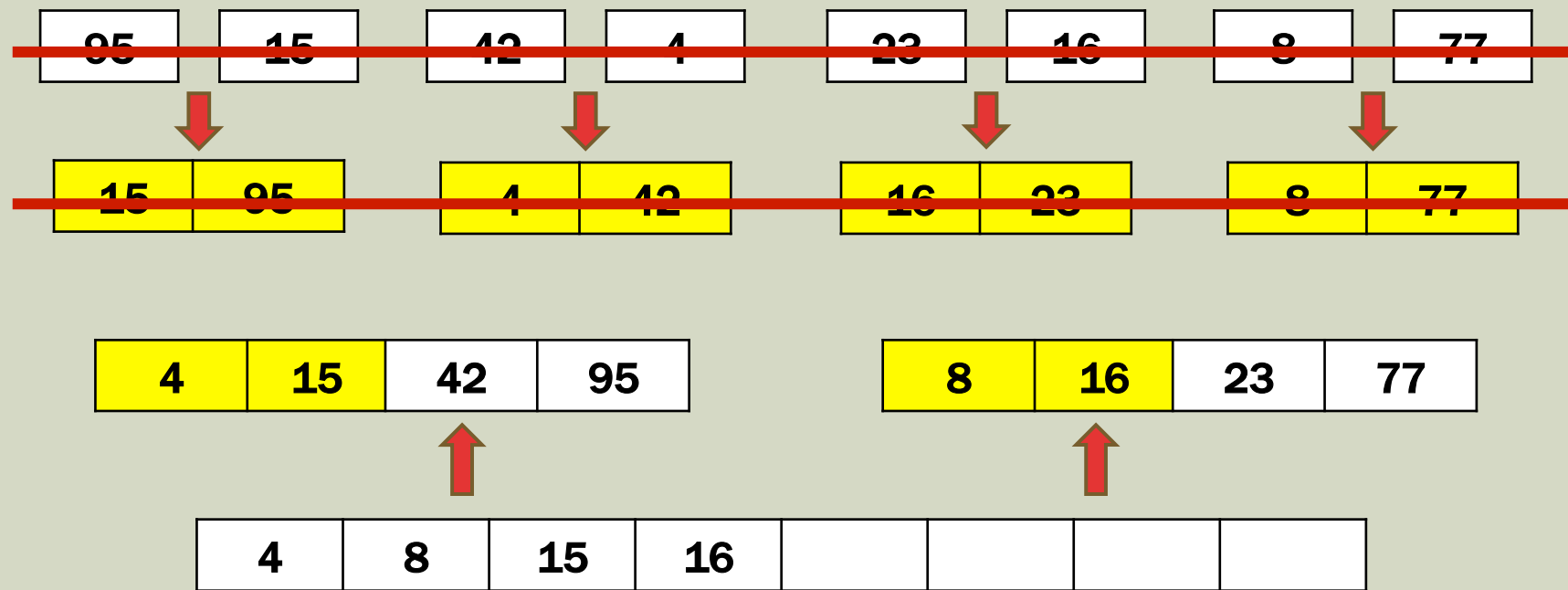
# MERGESORT



# MERGESORT

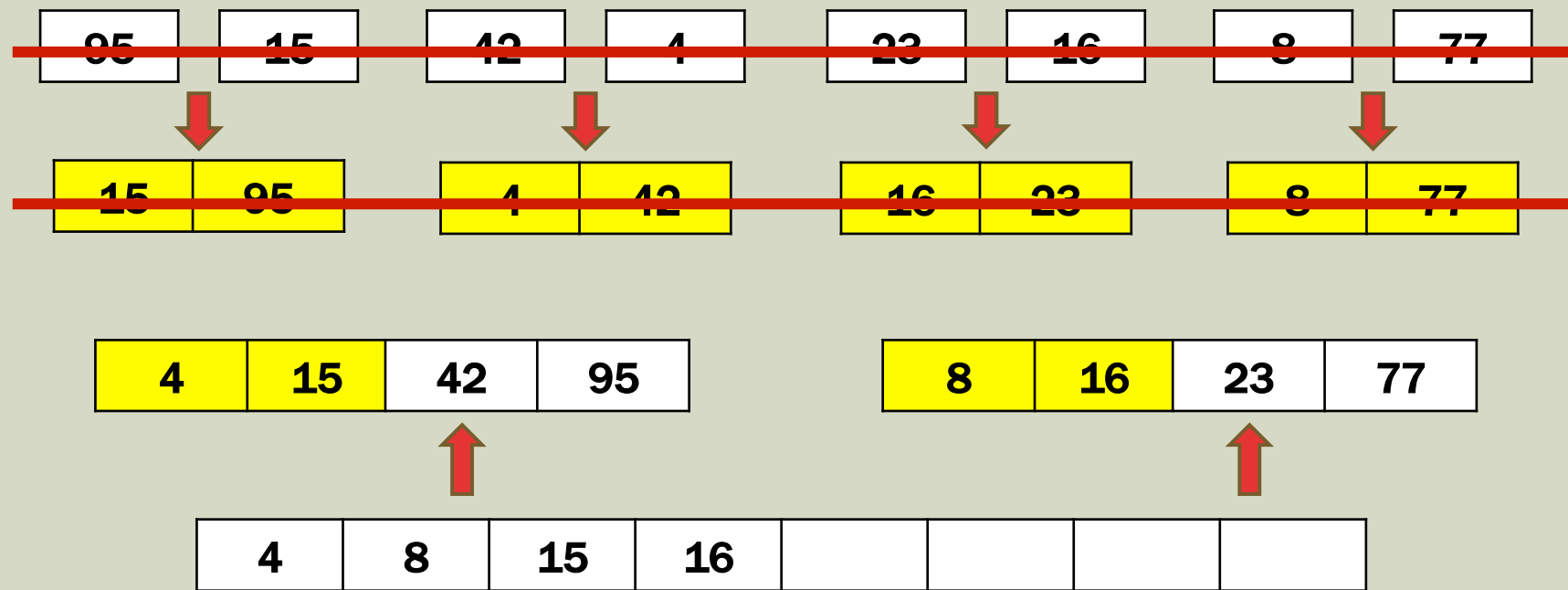


# MERGESORT

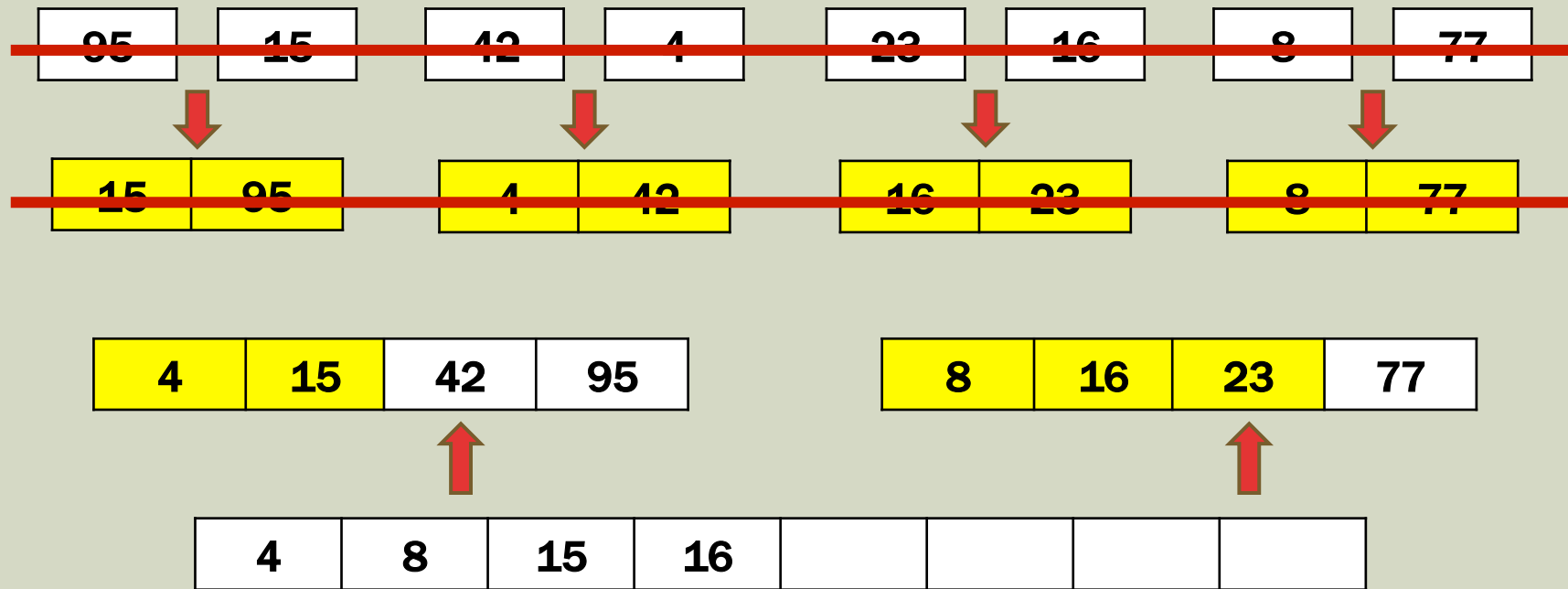




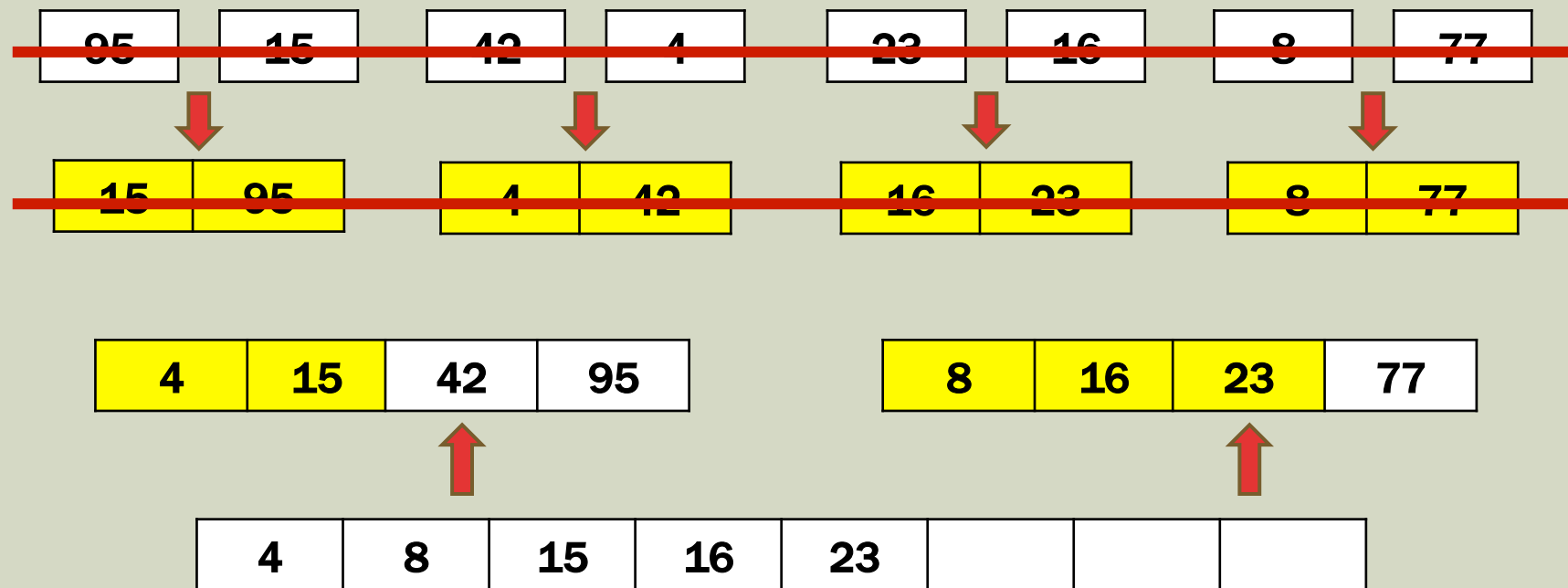
# MERGESORT



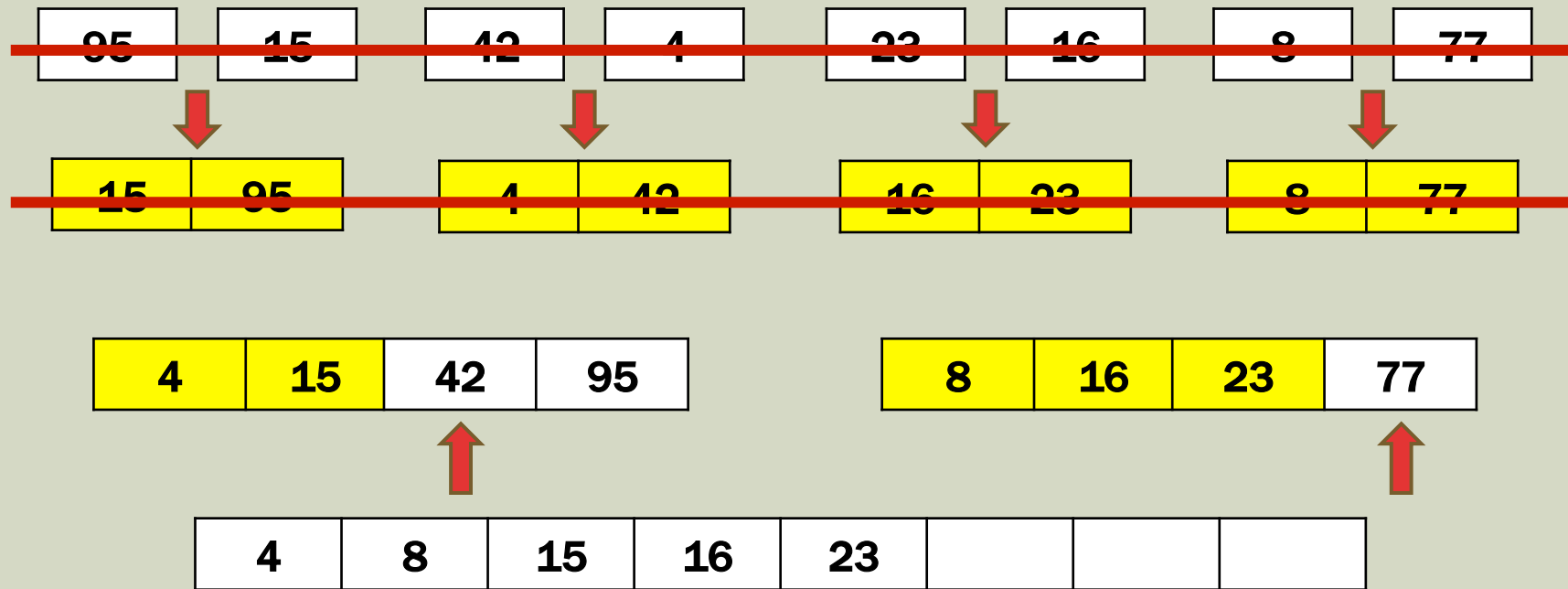
# MERGESORT



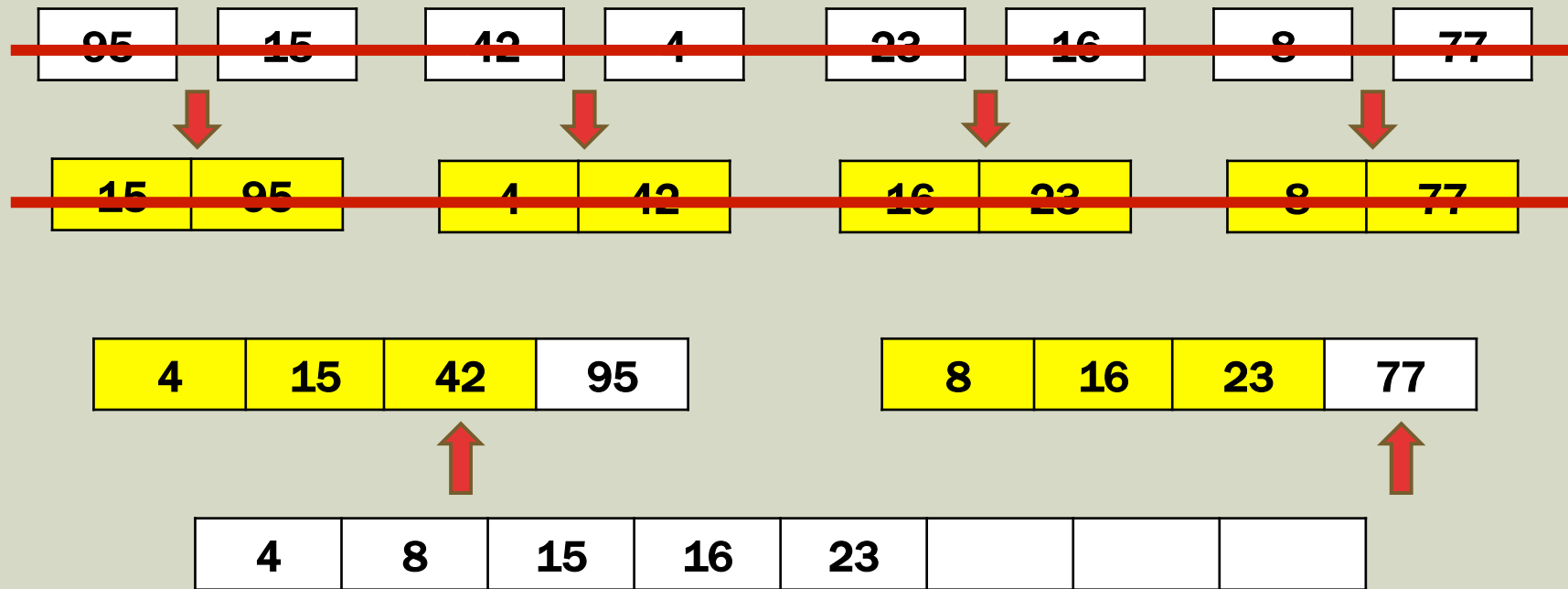
# MERGESORT



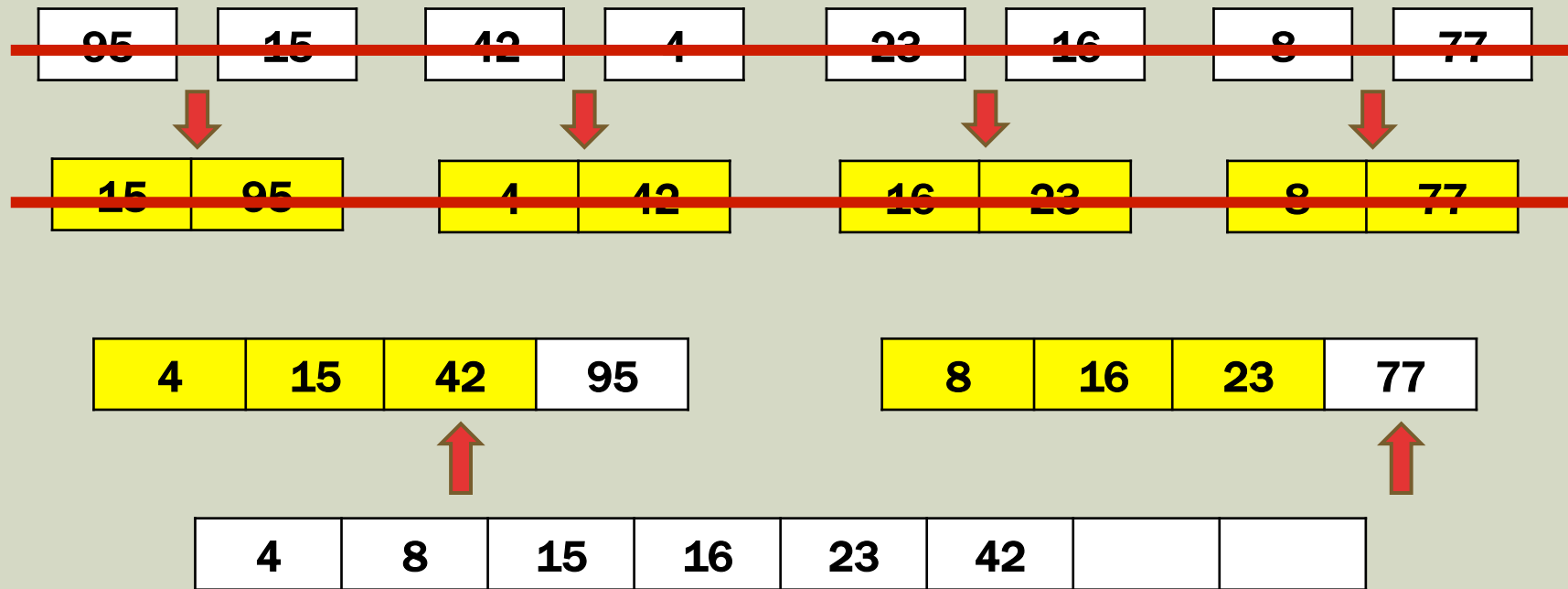
# MERGESORT



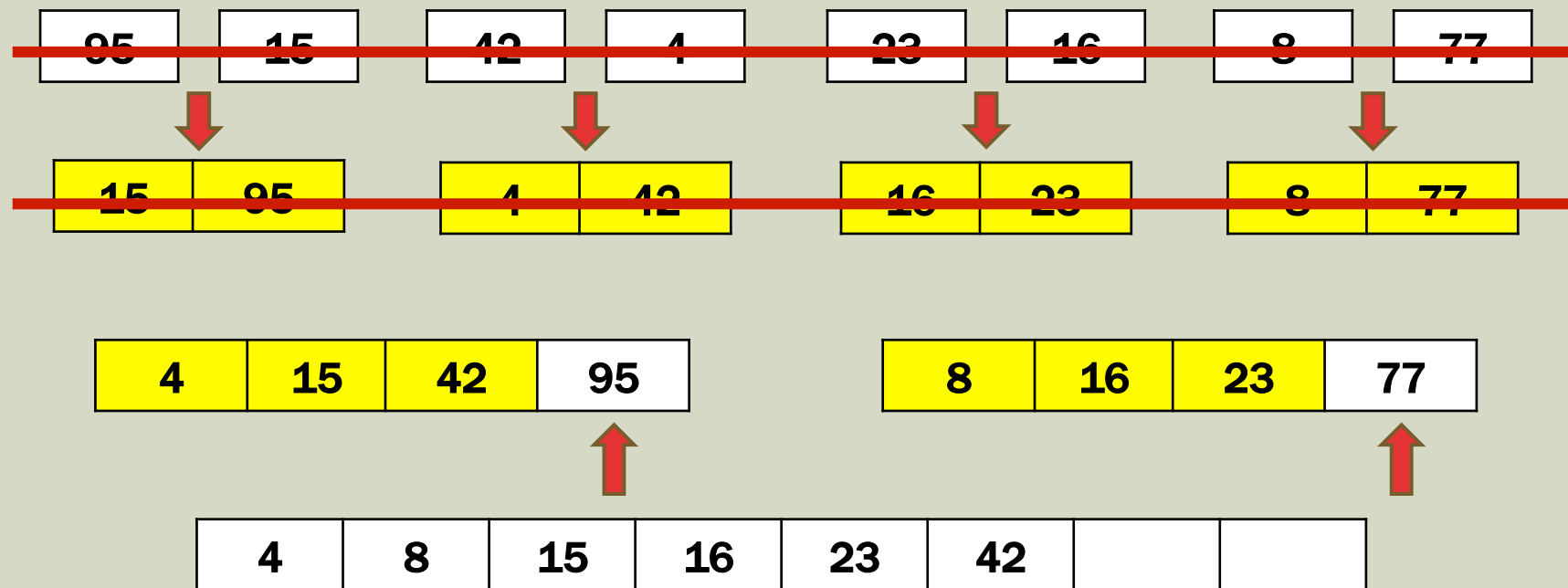
# MERGESORT



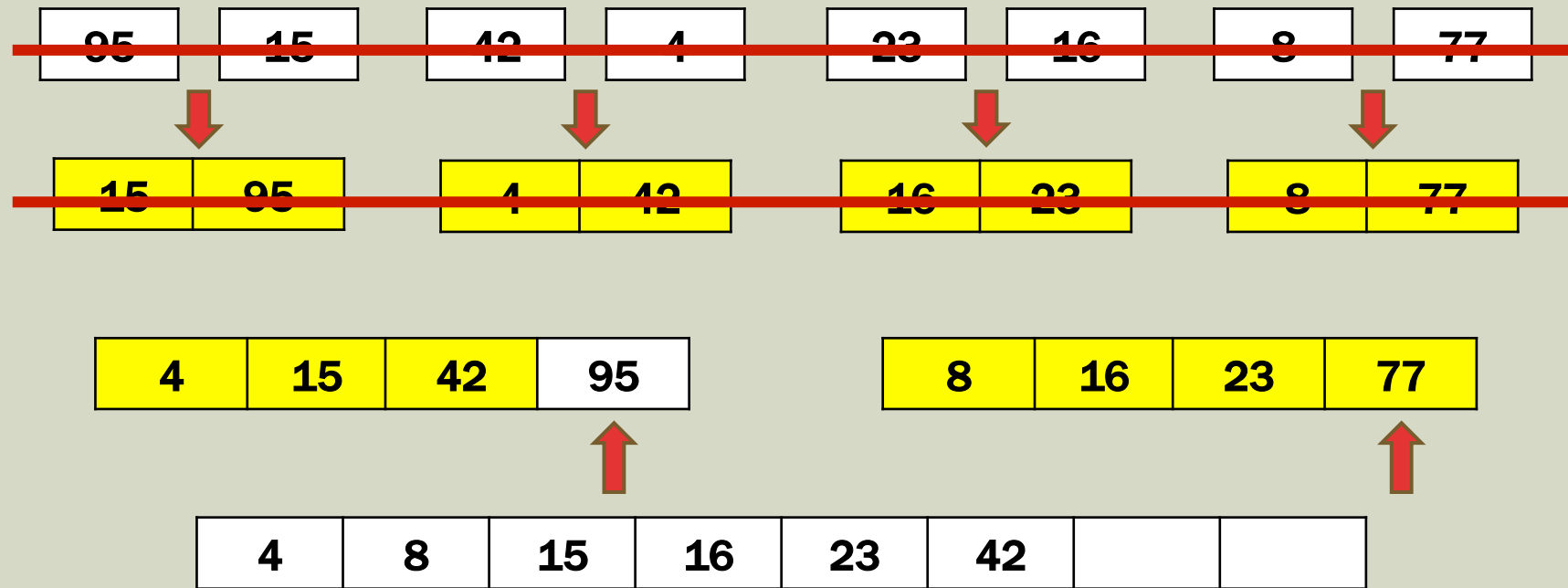
# MERGESORT



# MERGESORT

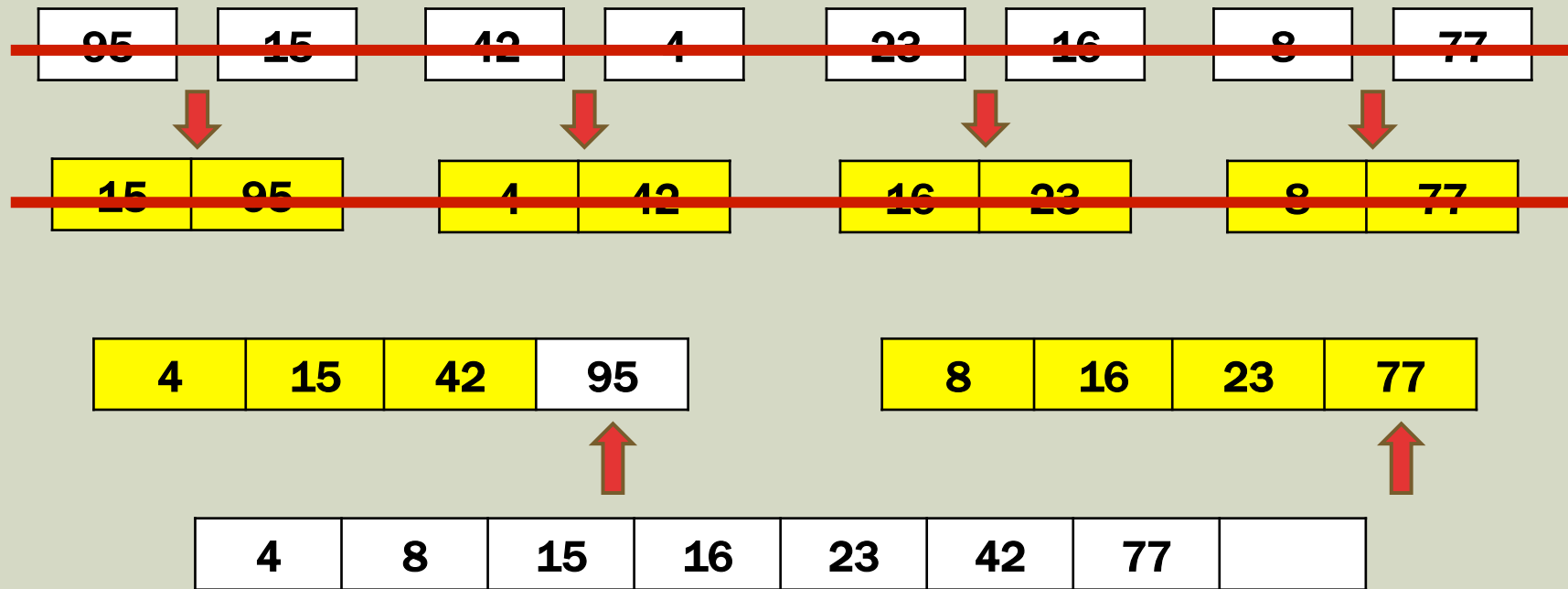


# MERGESORT

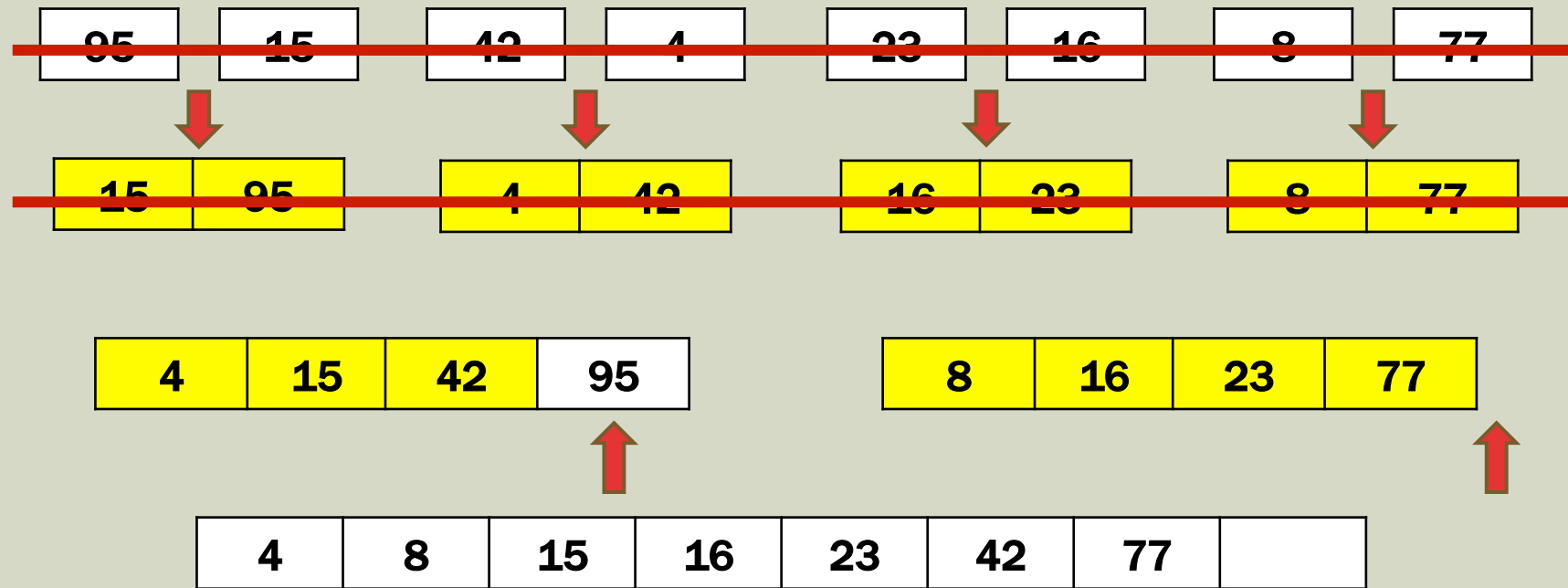




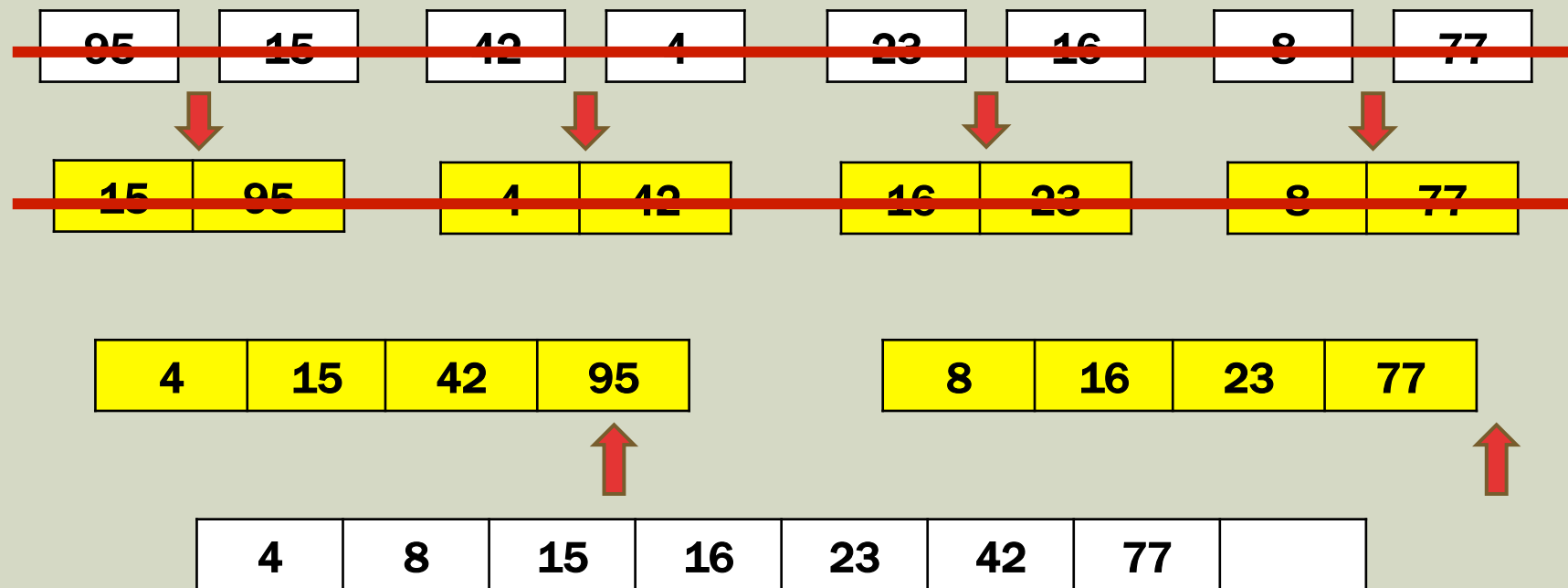
# MERGESORT



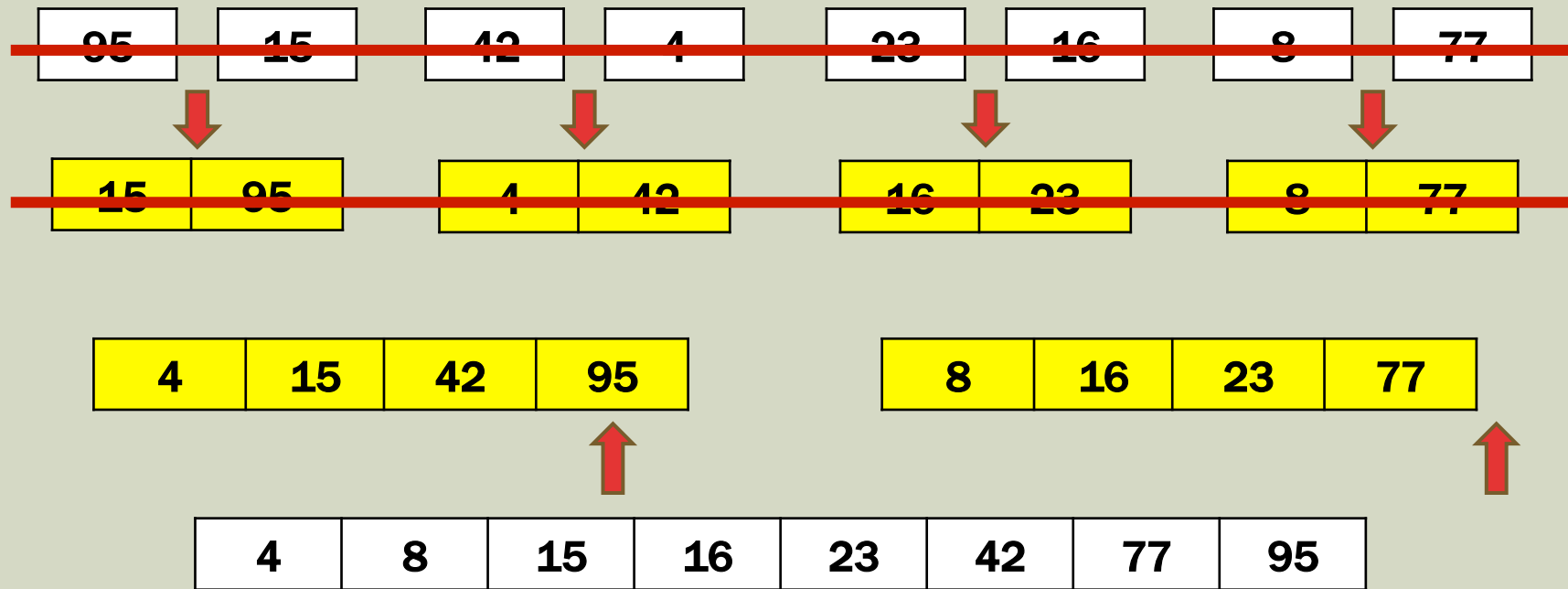
# MERGESORT



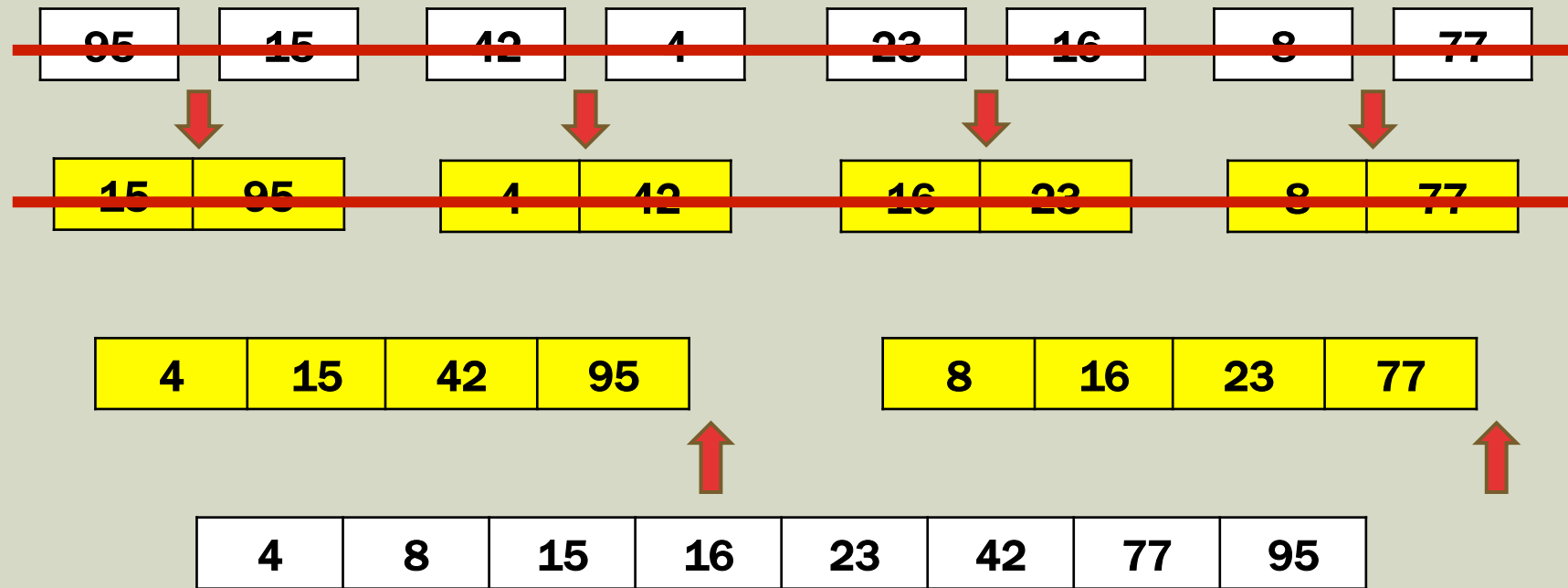
# MERGESORT



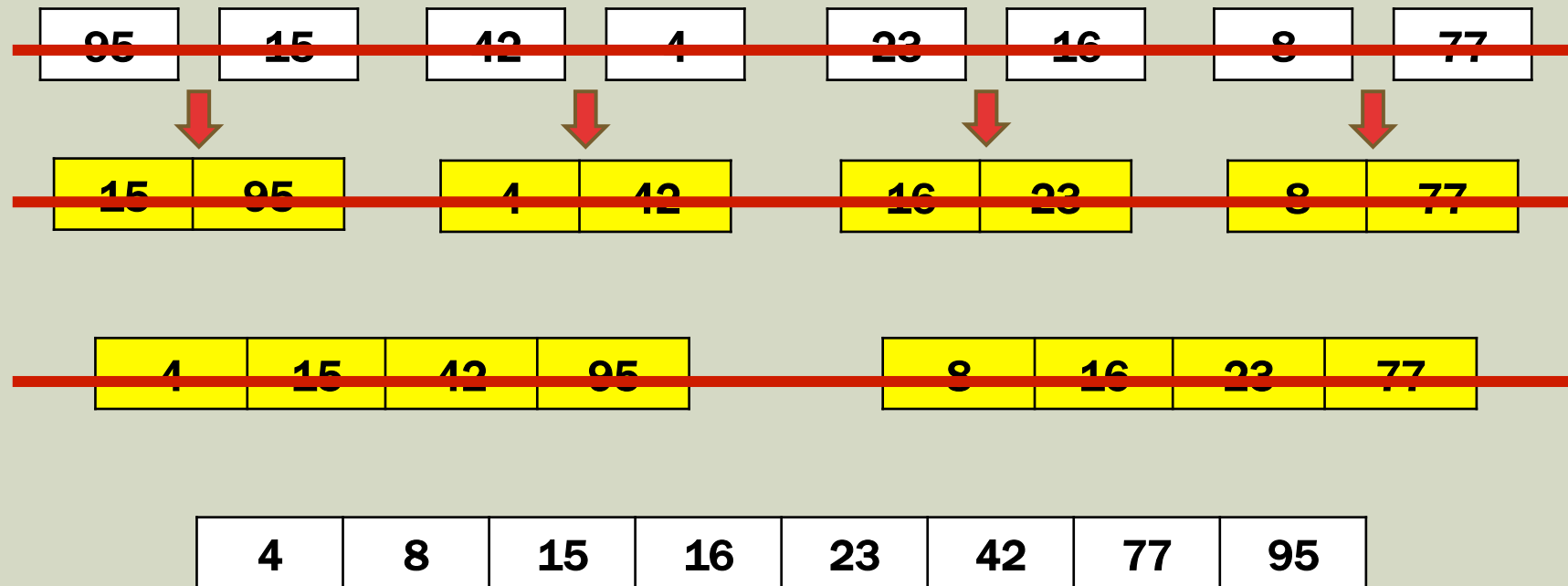
# MERGESORT



# MERGESORT



# MERGESORT



# ALGORITMO

```
void mergesort(vetor v, inteiro inicio, inteiro fim){  
    se (inicio < fim){  
        inteiro meio = (inicio + fim) / 2;  
        mergesort(v, inicio, meio);  
        mergesort(v, meio + 1, fim);  
        intercalar(v, inicio, fim);  
    }  
}
```

# ALGORITMO

- O “segredo” está no método *intercalar*.
- A intercalação percorre um subvetor;
  - comparando-o com o subvetor seguinte.



# CONSIDERAÇÕES

- Método estável.
- Necessita de espaço de memória adicional:
  - chamadas recursivas;
  - subvetores.