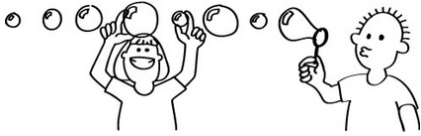


## Intercalação



Fonte: <http://csunplugged.org/sorting-algorithms>  
PF 9

<http://www.ime.usp.br/~pf/algoritmos/aulas/mrgsrt.html>

## Intercalação

**Problema:** Dados  $v[p : q]$  e  $v[q : r]$  crescentes, rearranjar  $v[p : r]$  de modo que ele fique em ordem crescente.

Para que valores de  $q$  o problema faz sentido?

Entra:

	$p$				$q$					$r$
v	22	33	55	77	11	44	66	88	99	

Sai:

	$p$				$q$					$r$
v	11	22	33	44	55	66	77	88	99	

## Intercalação

	$p$				$q$					$r$
v	22	33	55	77	11	44	66	88	99	

w										
---	--	--	--	--	--	--	--	--	--	--

## Intercalação

	$k$									
v										

	$i$									$j$
w	22	33	55	77	99	88	66	44	11	

## Intercalação

	$k$									
v	11									

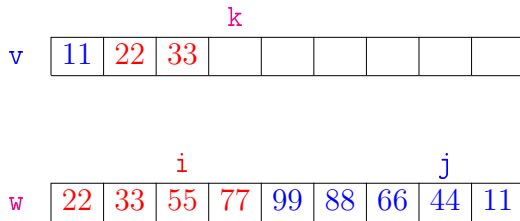
	$i$									$j$
w	22	33	55	77	99	88	66	44	11	

## Intercalação

	$k$									
v	11	22								

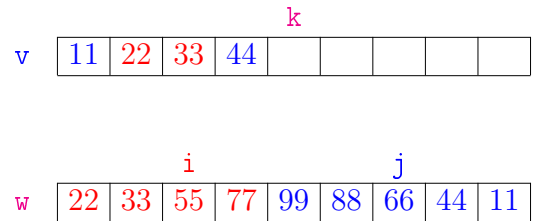
	$i$									$j$
w	22	33	55	77	99	88	66	44	11	

### Intercalação



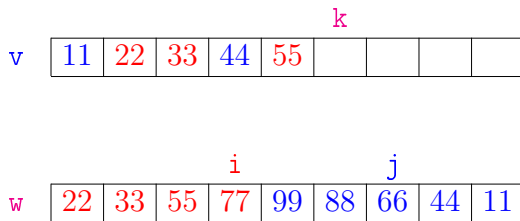
Navigation icons: back, forward, search, etc.

### Intercalação



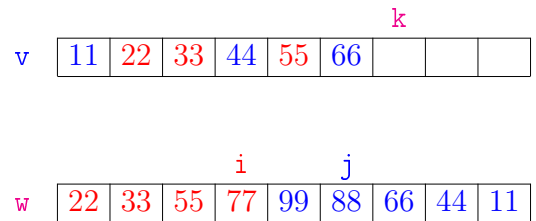
Navigation icons: back, forward, search, etc.

### Intercalação



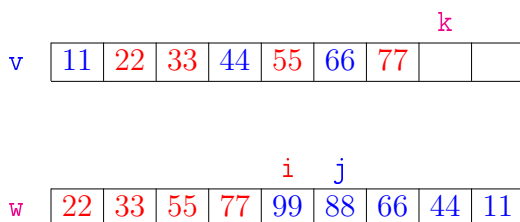
Navigation icons: back, forward, search, etc.

### Intercalação



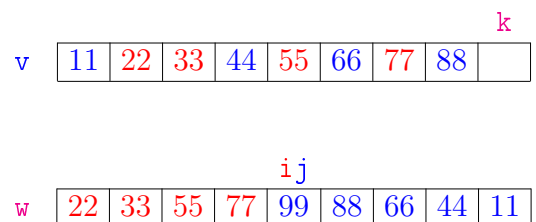
Navigation icons: back, forward, search, etc.

### Intercalação



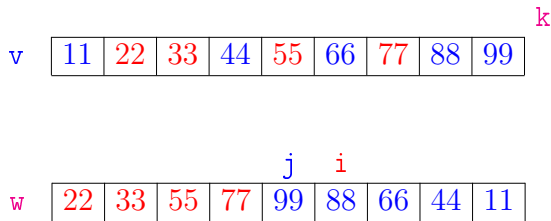
Navigation icons: back, forward, search, etc.

### Intercalação



Navigation icons: back, forward, search, etc.

## Intercalação



Navigation icons

## Intercalação

```
def intercale(p, q, r, v):
1  e = v[p:q] # clone
2  d = v[q:r] # clone
3  d.reverse() # método mutador
4  w = e + d
5  i = 0
6  j = r-p-1
7  for k in range(p,r):
8      if w[i] <= w[j]):
9          v[k] = w[i]
10         i += 1
11     else:
12         v[k] = w[j]
13         j -= 1
```

Navigation icons

## Consumo de tempo

$$n := r - p$$

linha	proporcional a
1-2	?
3	?
4	?
5-6	?
7	?
8-13	?
<b>total</b>	?

Navigation icons

## Consumo de tempo

$$n := r - p$$

linha	proporcional a
1-2	= n
3	≤ n
4	= n
5-6	= 1
7	= r - p + 1 = n + 1
8-13	= r - p = n
<b>total</b>	≈ 5n + 2

Navigation icons

## Conclusão

A função `intercale` consome  $5n + 2$  unidades de tempo.

O algoritmo `intercale` consome  $O(n)$  unidades de tempo.

Também escreve-se

O algoritmo `intercale` consome tempo  $O(n)$ .

Navigation icons