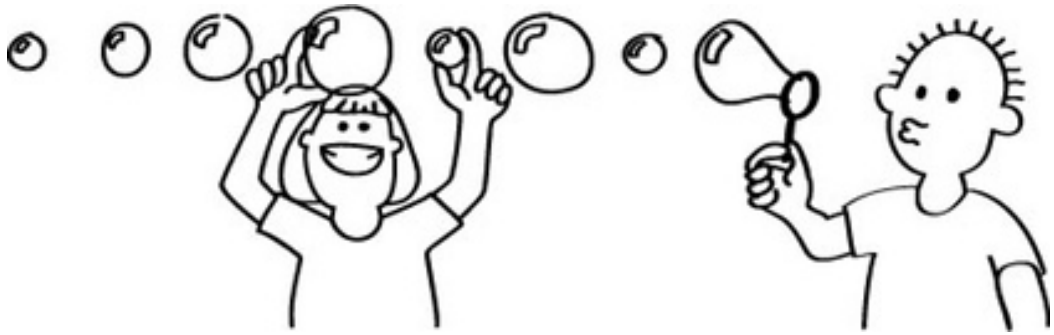
Intercalação



Fonte: http://csunplugged.org/sorting-algorithms

PF 9

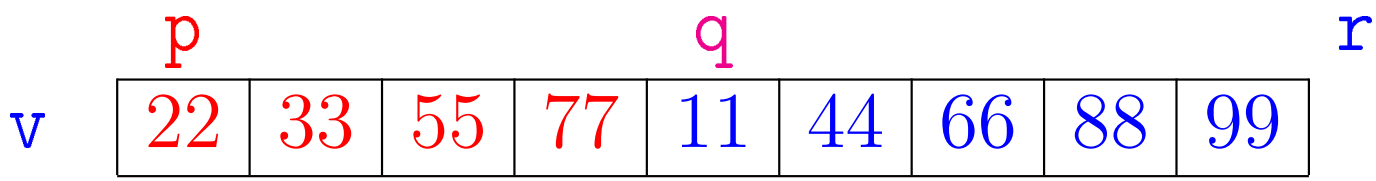
http://www.ime.usp.br/~pf/algoritmos/aulas/mrgsrt.html

Intercalação

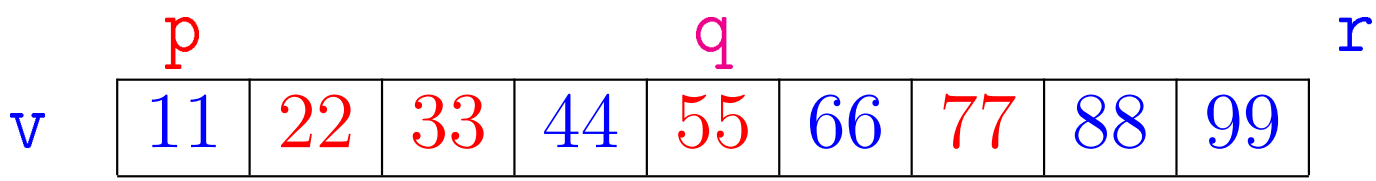
Problema: Dados v[p : q] e v[q : r] crescentes, rearranjar v[p : r] de modo que ele fique em ordem crescente.

Para que valores de q o problema faz sentido?

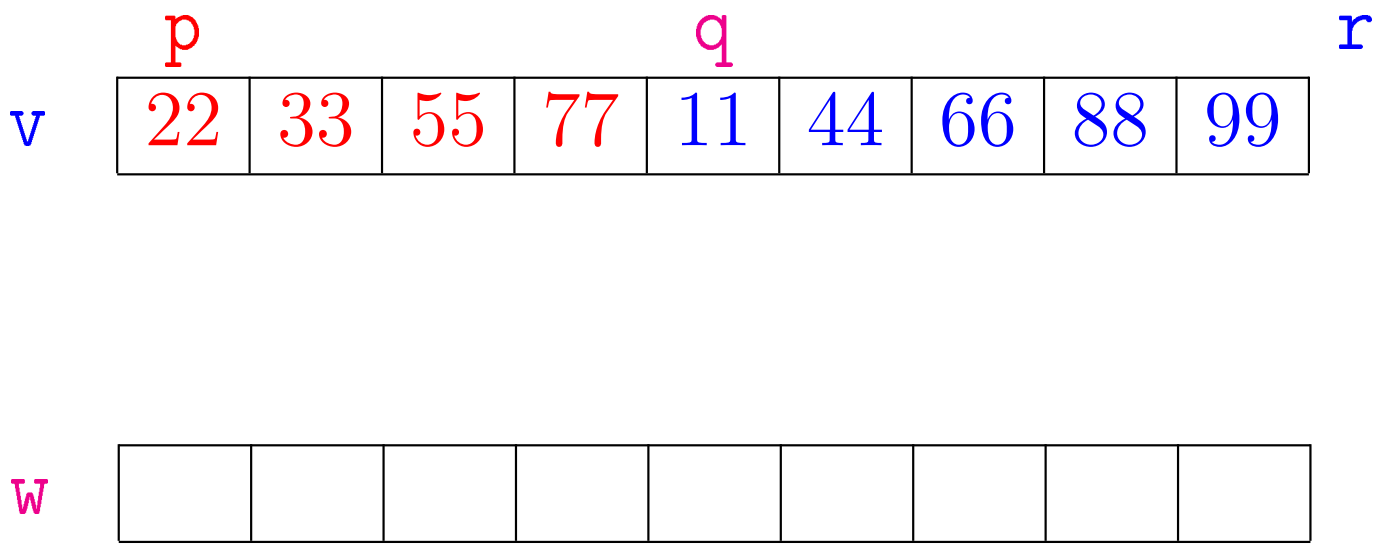
Entra:



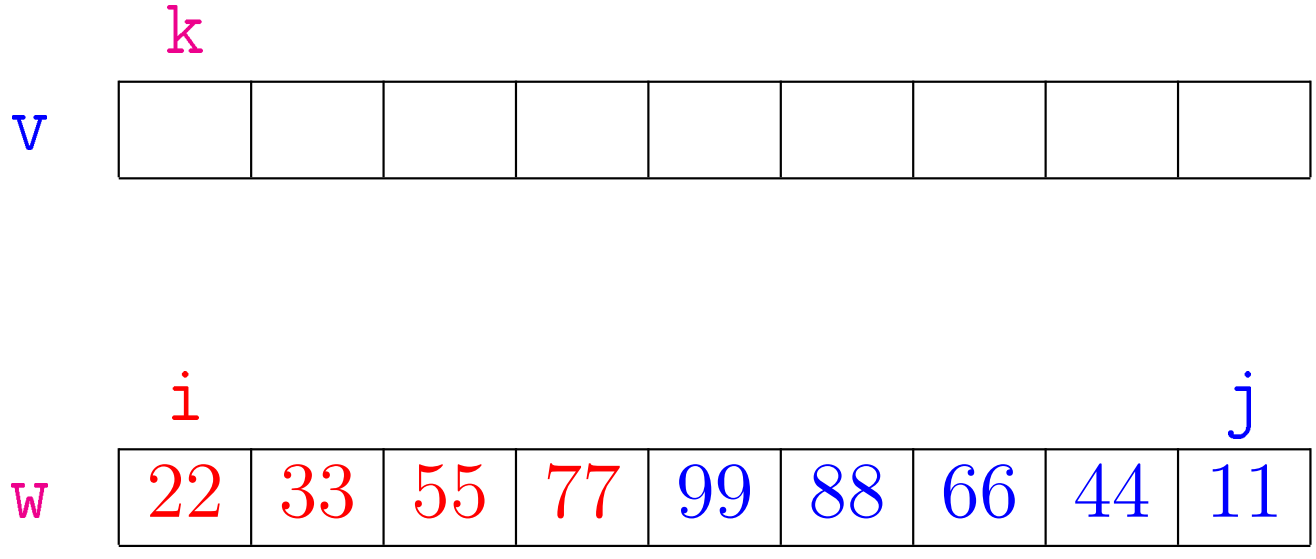
Sai:



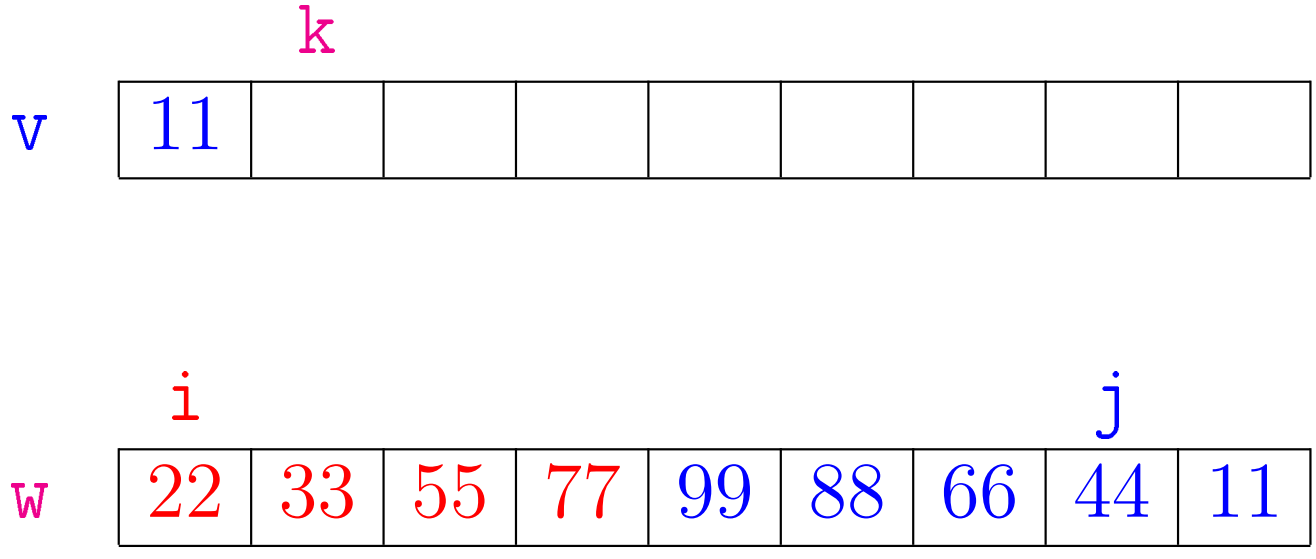
Intercalação



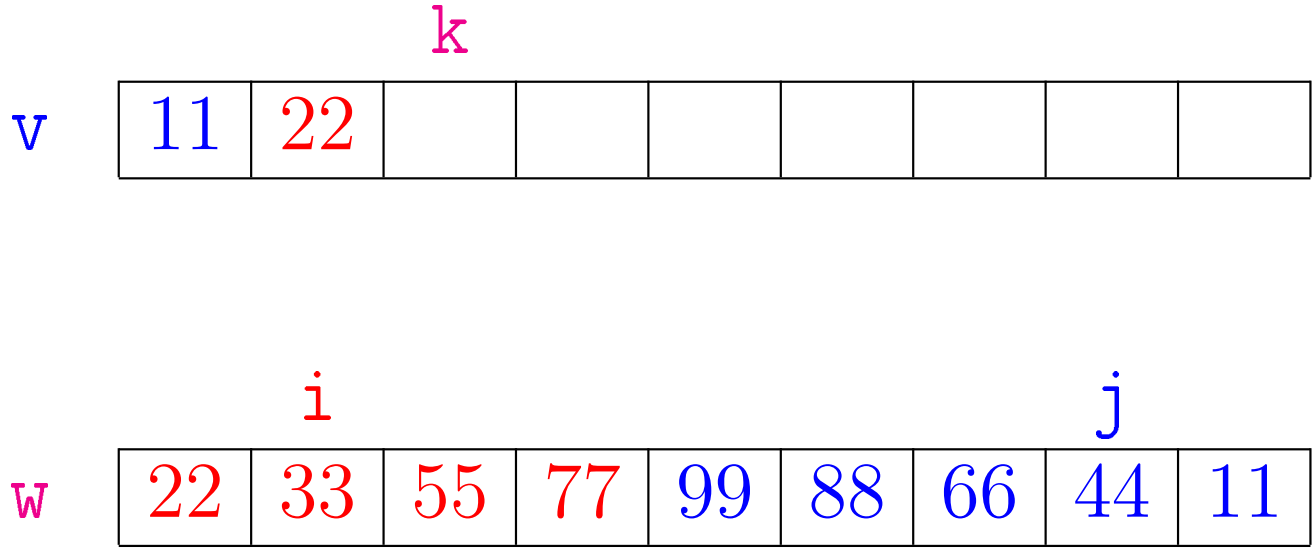
Intercalação



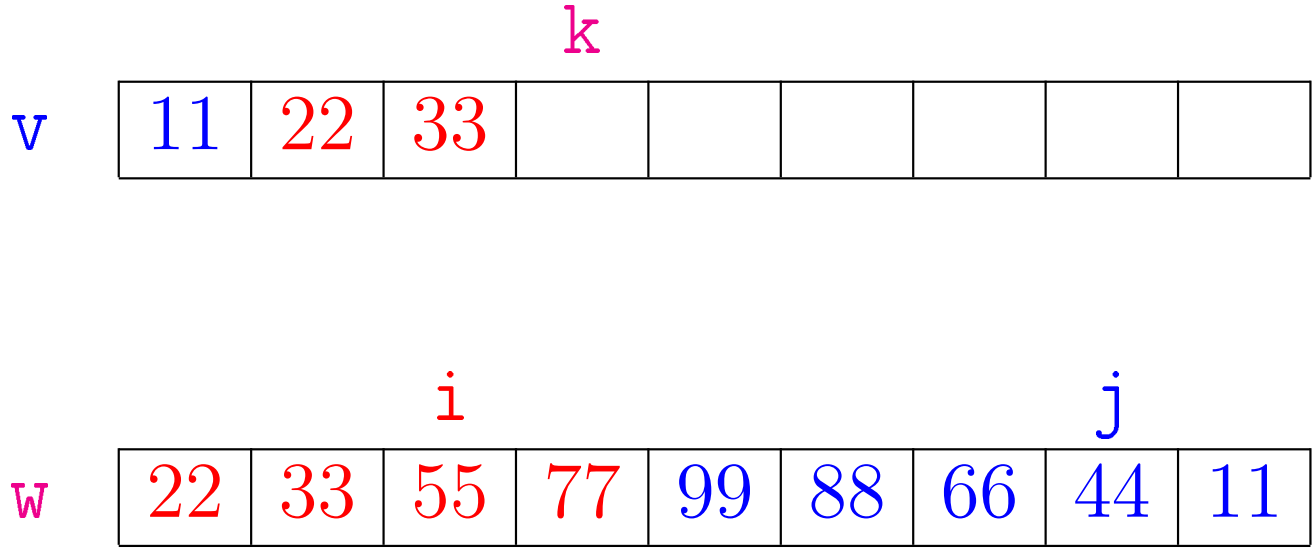
Intercalação



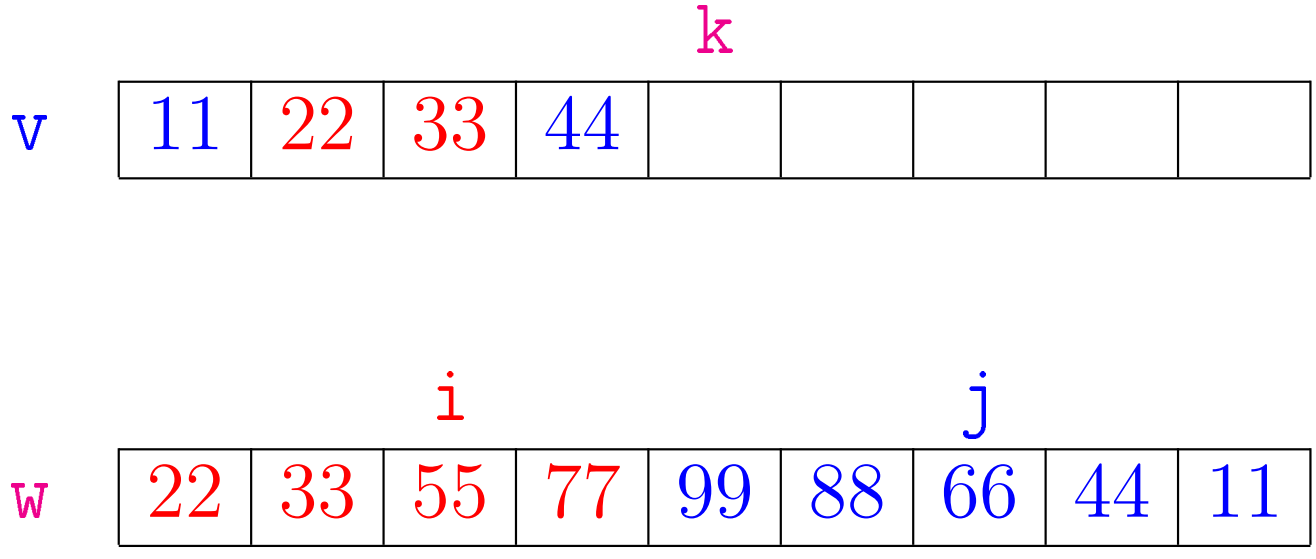
Intercalação



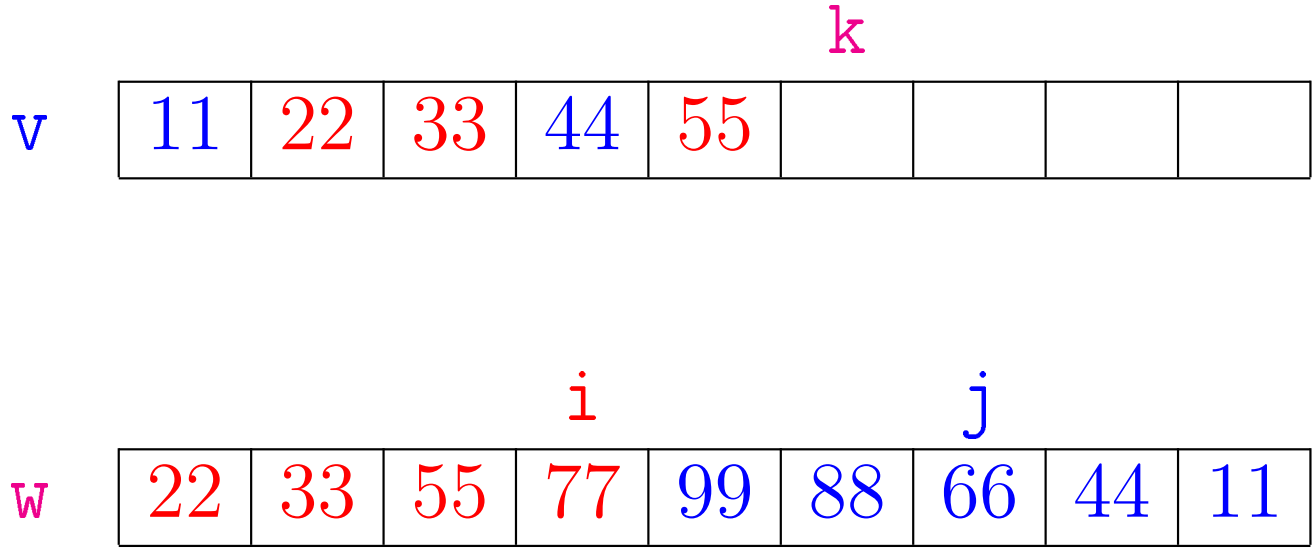
Intercalação



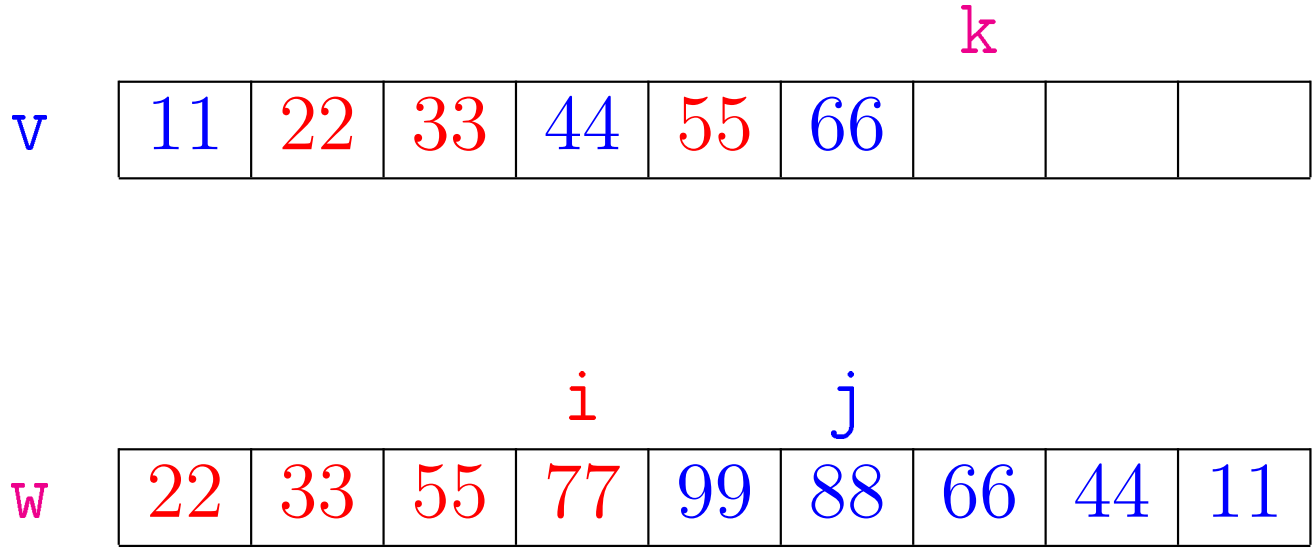
Intercalação



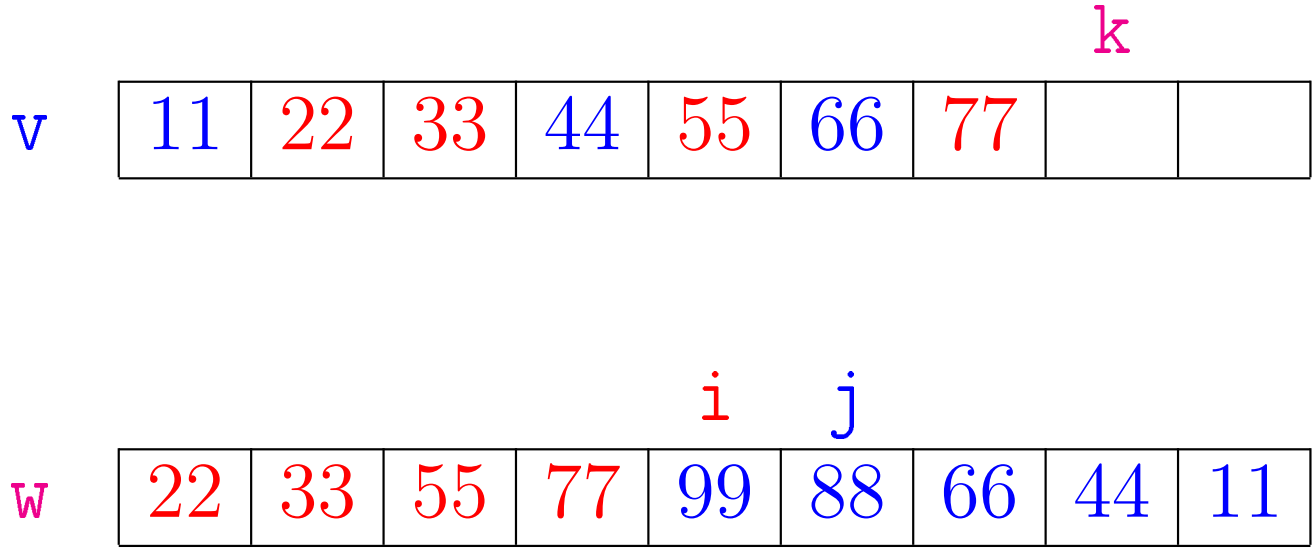
Intercalação



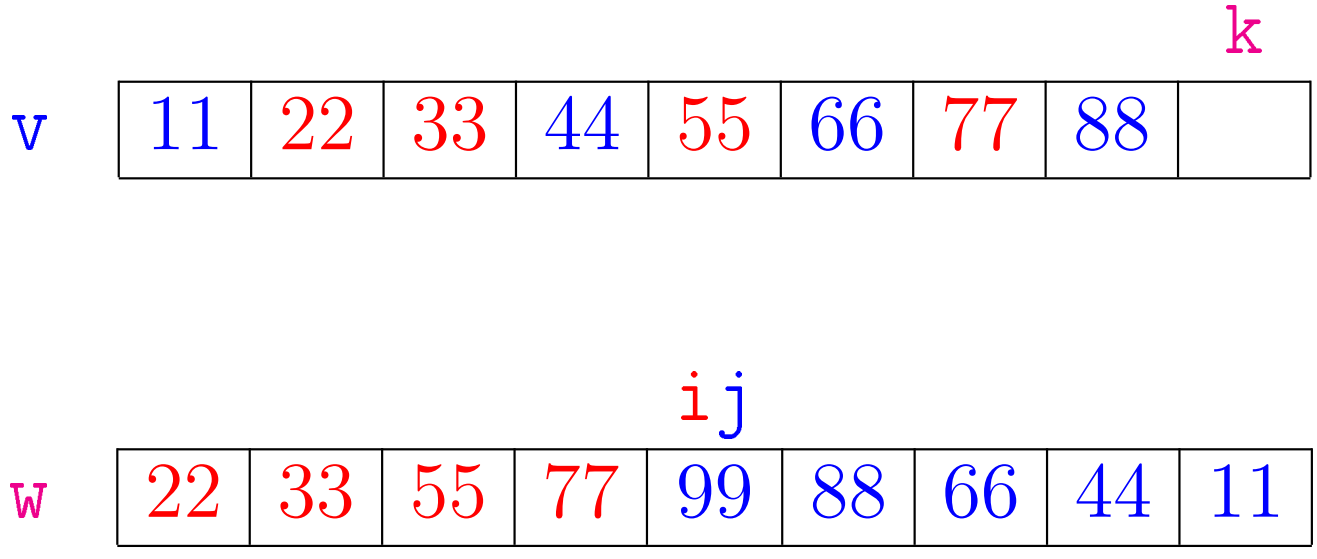
Intercalação



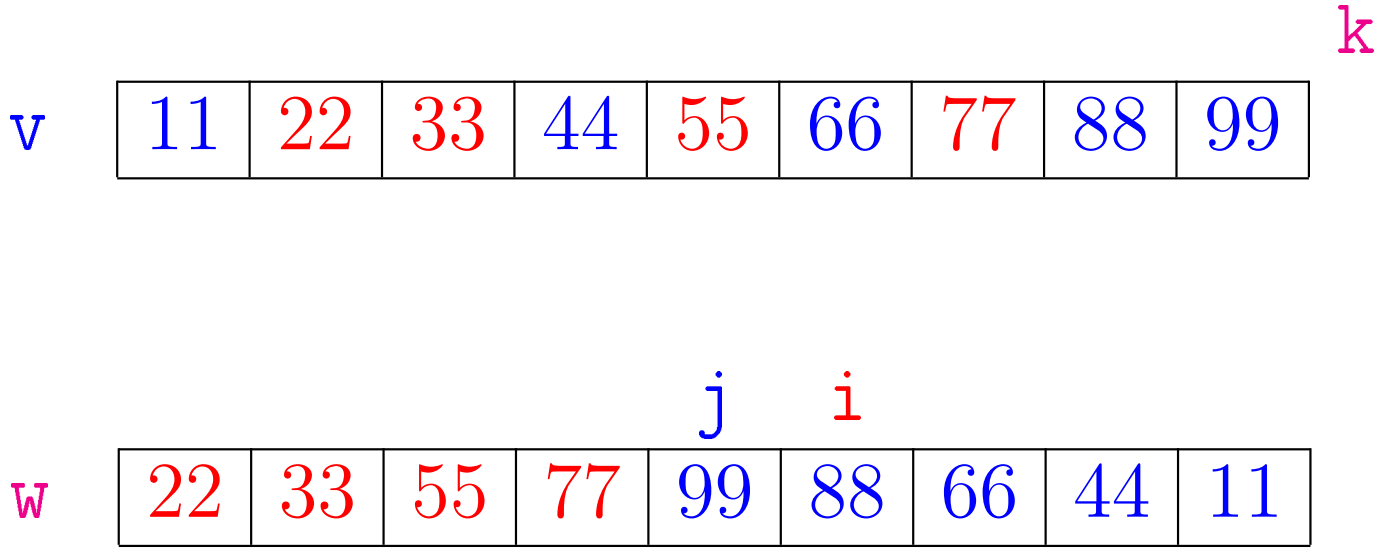
Intercalação



Intercalação



Intercalação



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
```

Consumo de tempo

`n := r - p`

linha	proporcional a
1–2	?
3	?
4	?
5–6	?
7	?
8–13	?
total	?

Consumo de tempo

$\mathtt{n} := \mathtt{r} - \mathtt{p}$

linha	proporcional a			
1–2	$=$	$\mathtt{n}$		
3	$\leq$	$\mathtt{n}$		
4	$=$	$\mathtt{n}$		
5–6	$=$	1		
7	$=$	$\mathtt{r} - \mathtt{p} + 1$	$=$	$\mathtt{n} + 1$
8–13	$=$	$\mathtt{r} - \mathtt{p}$	$=$	$\mathtt{n}$
total	$\approx$	$5\mathtt{n} + 2$		

Conclusão

A função `intercale` consome $5\mathtt{n} + 2$ unidades de tempo.

O algoritmo `intercale` consome $\mathrm{O}(\mathtt{n})$ unidades de tempo.

Também escreve-se

O algoritmo `intercale` consome tempo $\mathrm{O}(\mathtt{n})$.