

- I. $v1 = \{7, 5, 5, 3\}$
 $v2 = \{1, 2, 3, 4\}$
 $v3 = \{4, 3, 2, 1\}$

Passo 1:

$v1 = \{7, 5, 5, 3\}$
 $v2 = \{1, 2, 3, 4\}$
 $v3 = \{4, 3, 2, 1\}$

Passo 2:

$v1 = \{7, 5, 5, 3\}$
 $v2 = \{1, 2, 3, 4\}$
 $v3 = \{1, 3, 2, 1\}$

Passo 3:

$v1 = \{7, 5, 5, 3\}$
 $v2 = \{1, 2, 3, 4\}$
 $v3 = \{1, 2, 2, 1\}$

Passo 4:

$v1 = \{7, 5, 5, 3\}$
 $v2 = \{1, 2, 3, 4\}$
 $v3 = \{1, 2, 3, 1\}$

Passo 5:

$v1 = \{7, 5, 5, 3\}$
 $v2 = \{1, 2, 3, 4\}$
 $v3 = \{1, 2, 3, 4\}$

Resultado final: $v3 = \{1, 2, 3, 4\}$

- II. A quantidade de trocas no merge é proporcional ao tamanho total dos vetores que estão sendo mesclados. Cada elemento de $v2$ será comparado e movido para a posição correta em $v3$. Portanto, para $v2$, haverá um total de 4 trocas.
- III. Para ordenar em ordem decrescente, as alterações necessárias nas funções `merge()` e `intercala()` são simplesmente inverter a comparação. Ou seja, onde há uma comparação do tipo `if A[i] <= B[j]`, ela seria substituída por `if A[i] >= B[j]`. Além disso, se houver alguma troca de elementos, ela também deve ser ajustada para garantir a ordem decrescente.
- IV. $v = \{2, 3, 4, 5, 6, 6, 7, 8, 9\}$