

Imagine que nós temos duas listas U e W , ambas com tamanho n .
As duas listas contém os mesmos elementos, em ordens diferentes.
Por exemplo,

U	7	25	11	40	1	8	14	16	10	9
W	40	8	10	14	16	9	7	1	25	11

Mas daí, pode acontecer que para uma tripla qualquer (x, y, z) , os três números aparecem na mesma ordem nas duas listas.
Quando isso acontece, nós dizemos que (x, y, z) é uma tripla engraçada.
No exemplo acima, $(8, 16, 9)$ é uma tripla engraçada

U	7	25	11	40	1	8	14	16	10	9
W	40	8	10	14	16	9	7	1	25	11

Mas a tripla $(7, 1, 10)$ não é

U	7	25	11	40	1	8	14	16	10	9
W	40	8	10	14	16	9	7	1	25	11

A nossa tarefa de hoje é
→ Contar o número total de triplas engraçadas em U e W

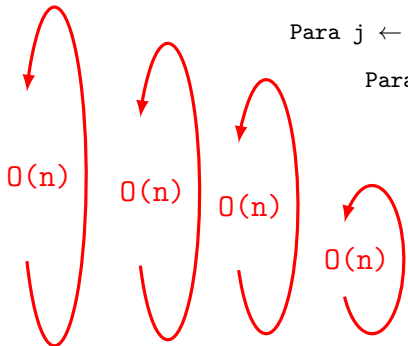
Mas, isso é uma coisa muito fácil.
Basta considerar toda tupla de elementos em U e verificar se eles aparecem na mesma ordem em W .
A coisa fica assim

```
func contaTE (U,W)

    cont ← 0

    Para i ← 2 Até N-2
        Para j ← i+1 Até N-1
            Para k ← j+1 Até N
                aux ← 1
                Para ℓ ← 1 Até N
                    Se ( aux = 1 E W[ℓ] = U[i] )      aux++
                    Sse ( aux = 2 E W[ℓ] = U[j] )      aux++
                    Sse ( aux = 3 E W[ℓ] = U[k] )
                        cont++;  Quebra

    Retorna (cont)
```



O problema é que esse algoritmo é uma grande porcaria: $O(n^4)$.