



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Comparación de preferencias

Análisis y Diseño de Algoritmos

Comparación de preferencias

Un sitio web intenta comparar las preferencias de un usuario con las de otros:

- El usuario establece un ranking de n productos.
- La aplicación web consulta su base de datos para encontrar usuarios con gustos similares.

Medida de similitud:

Número de "inversiones" entre dos rankings:

- Ranking A: $1, 2, \dots, n$.
- Ranking B: a_1, a_2, \dots, a_n .
- i y j están invertidos si $i < j$ pero $a_i > a_j$.



Comparación de preferencias



Productos

	P1	P2	P3	P4	P5
A	1	2	3	4	5
B	1	3	4	2	5

Inversiones
3-2, 4-2

Algoritmo por fuerza bruta:
Comprobar todos los pares (i,j)
 $\Theta(n^2)$



Comparación de preferencias



Algoritmo "divide y vencerás"

- **División:**
Dividir la lista de productos en dos mitades y contar recursivamente el número de inversiones en cada mitad.
- **Combinación:**
Contar las inversiones en las que a_i y a_j están en mitades diferentes y devolver la suma de 3 cantidades.



Comparación de preferencias



Algoritmo "divide y vencerás"

1	5	4	8	10	2	6	9	12	11	3	7
---	---	---	---	----	---	---	---	----	----	---	---

1	5	4	8	10	2	6	9	12	11	3	7
---	---	---	---	----	---	---	---	----	----	---	---

5 inversiones

8 inversiones

División: $2T(n/2)$

9 inversiones entre una mitad y la otra:
5-3, 4-3, 8-6, 8-3, 8-7, 10-6, 10-9, 10-3, 10-7

Combinación: ???

Total = $5 + 8 + 9 = 22$.



Comparación de preferencias



Algoritmo "divide y vencerás"

Combinación

- Asumiendo que cada mitad está ordenada, se contabilizan las inversiones en las que a_i y a_j están en mitades diferentes.
- A continuación, se mezclan las dos mitades para devolver un conjunto ordenado.



Comparación de preferencias



Algoritmo "divide y vencerás"

Combinación



13 inversiones: $6 + 3 + 2 + 2 + 0 + 0$

Conteo: $O(n)$



Mezcla: $O(n)$

$$T(n) \leq T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + O(n) \Rightarrow T(n) = O(n \log n)$$



Comparación de preferencias



Algoritmo "divide y vencerás"

Implementación

```
Sort-and-Count(L)
{
  if (L.length==1)
    return (0, L);

  Dividir la lista en dos mitades A y B
  (rA, A) ← Sort-and-Count(A)
  (rB, B) ← Sort-and-Count(B)
  (r, L) ← Merge-and-Count(A, B)

  return (rA + rB + r, L);
}
```



Comparación de preferencias



Merge & Count

$i = 6$



3	7	10	14	18	19
---	---	----	----	----	----



2	11	16	17	23	25
---	----	----	----	----	----

--	--	--	--	--	--	--	--	--	--	--	--

Total:



Comparación de preferencias



Merge & Count

$i = 6$



3	7	10	14	18	19
---	---	----	----	----	----



2	11	16	17	23	25
---	----	----	----	----	----

6

2											
---	--	--	--	--	--	--	--	--	--	--	--

Total: 6

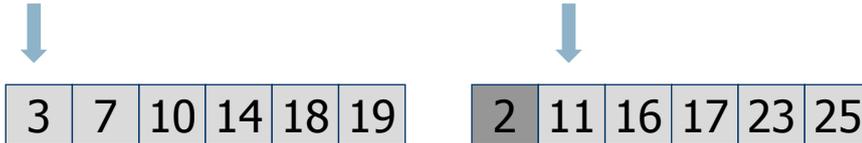


Comparación de preferencias



Merge & Count

$i = 6$



6



Total: 6

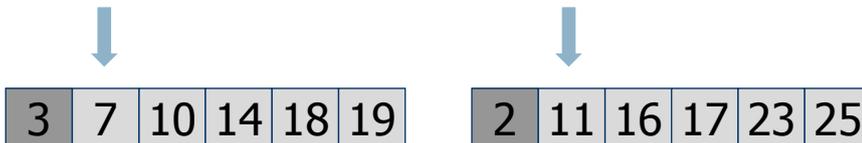


Comparación de preferencias



Merge & Count

$i = 5$



6



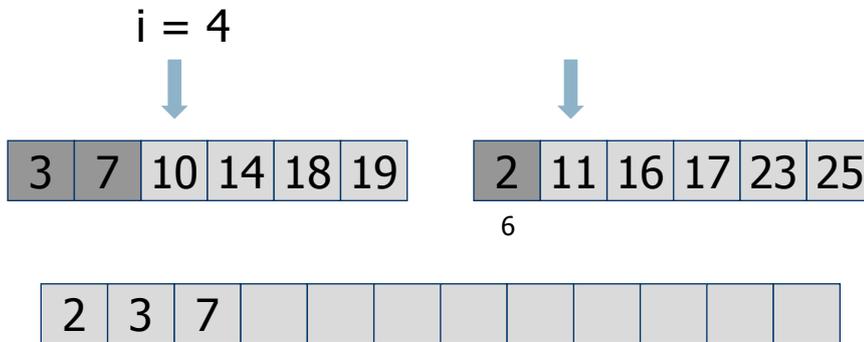
Total: 6



Comparación de preferencias



Merge & Count



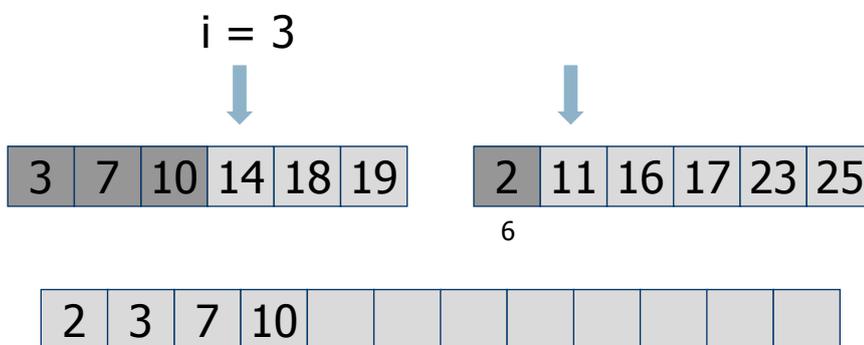
Total: 6



Comparación de preferencias



Merge & Count



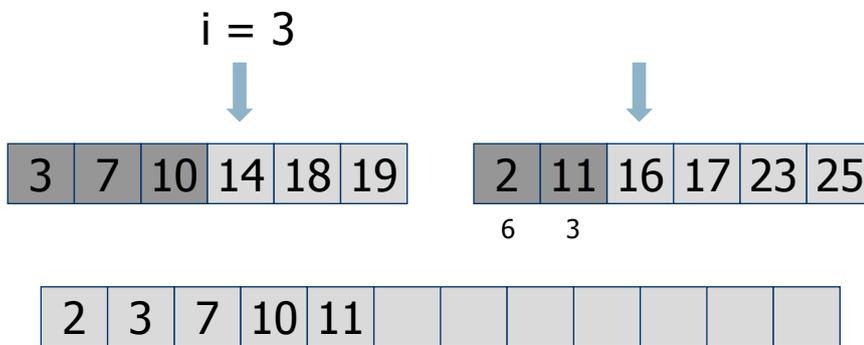
Total: 6



Comparación de preferencias



Merge & Count



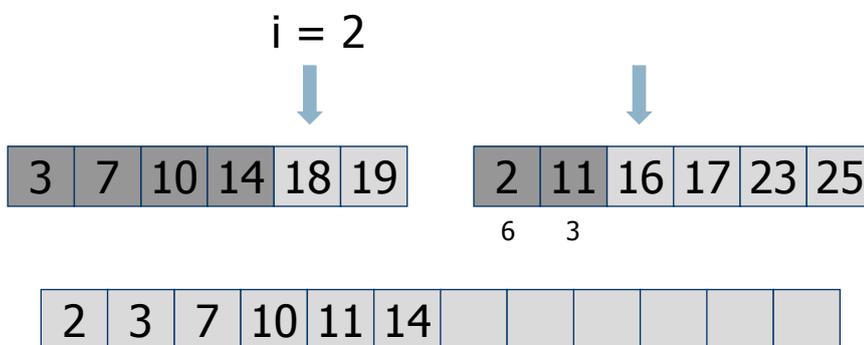
Total: 6 + 3



Comparación de preferencias



Merge & Count



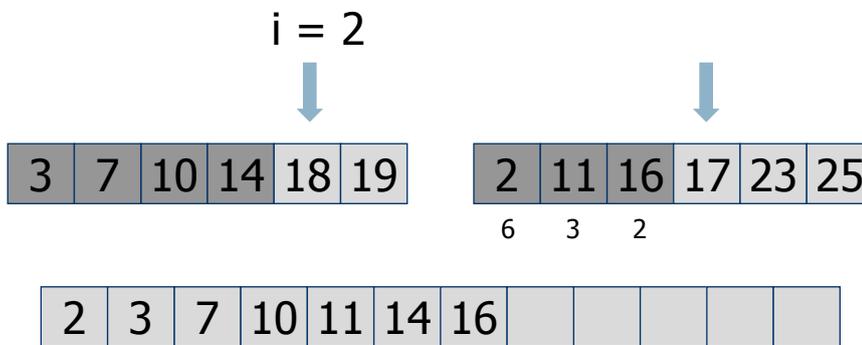
Total: 6 + 3



Comparación de preferencias



Merge & Count



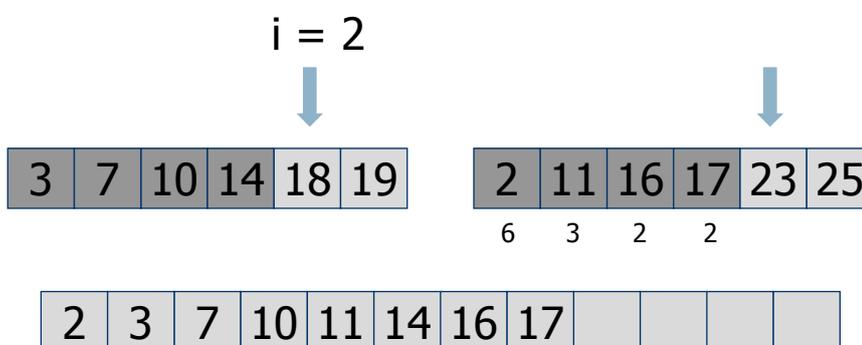
Total: $6 + 3 + 2$



Comparación de preferencias



Merge & Count



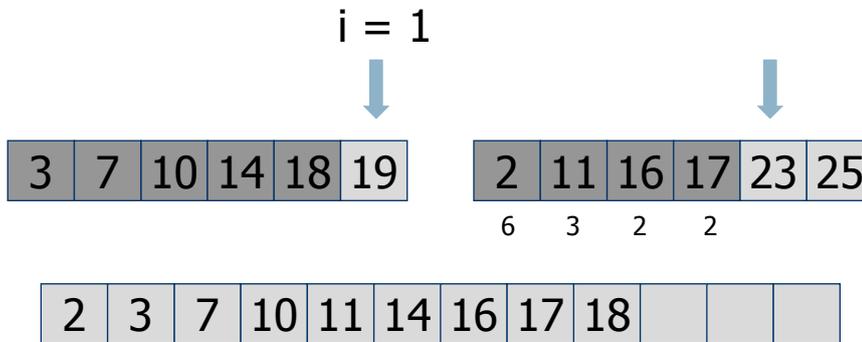
Total: $6 + 3 + 2 + 2$



Comparación de preferencias



Merge & Count



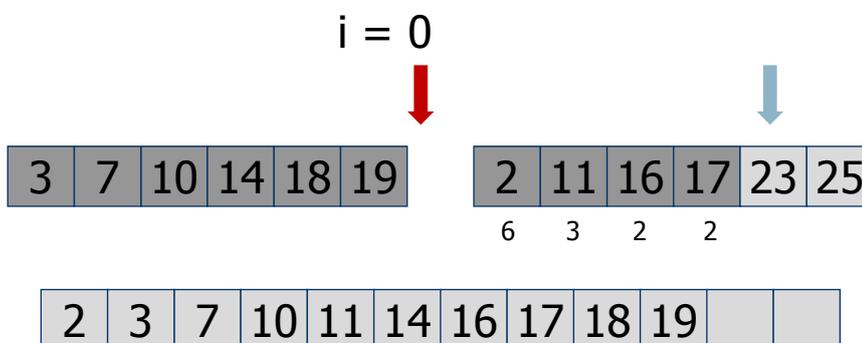
Total: $6 + 3 + 2 + 2$



Comparación de preferencias



Merge & Count



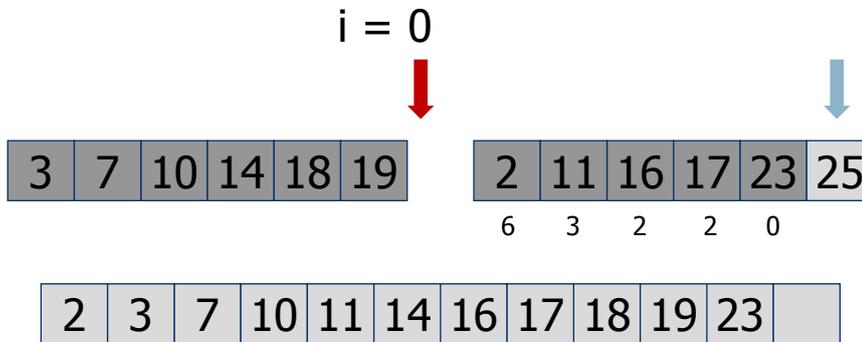
Total: $6 + 3 + 2 + 2$



Comparación de preferencias



Merge & Count



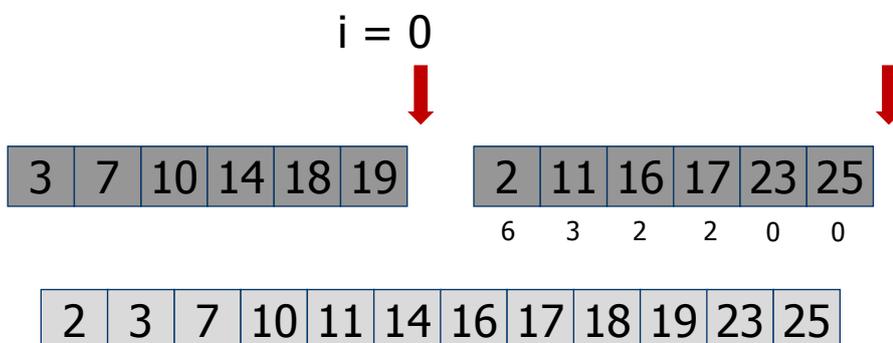
Total: $6 + 3 + 2 + 2 + 0$



Comparación de preferencias



Merge & Count



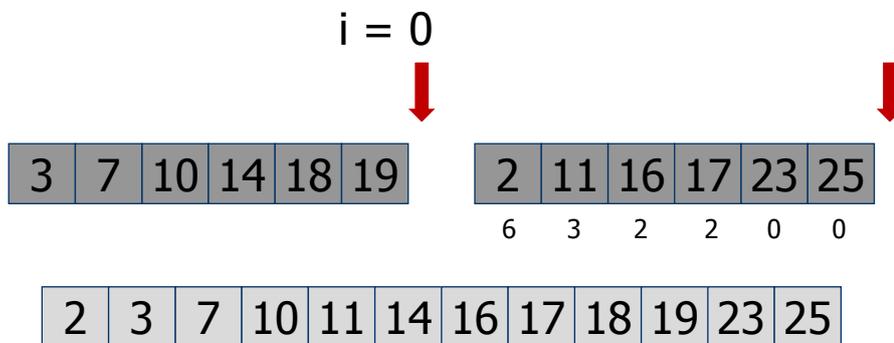
Total: $6 + 3 + 2 + 2 + 0 + 0$



Comparación de preferencias



Merge & Count



Total: $6 + 3 + 2 + 2 + 0 + 0 = \mathbf{13}$

