División y conquista: Contando inversiones

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Problema

Sean

Un conjunto de n “elementos”

Dos listas ordenadas de los “n” elementos

Queremos

Tener una medida de semejanza / diferencia entre las dos listas.

Diferencias entre las preferencias

Llamaremos

A y B a las listas de preferencias.

Utilizaremos

El orden de aparición de los elementos de A para identificar a los diferencias.

A=1,2,. . .,n

Por otro lado

Tendremos la lista de B que posiblemente esté “fuera de orden”

Ej: B=2,1,4,7,n,. . .

(Si B esta en orden, entonces representa el mismo orden de preferencia que A)

Mensurar las diferencias

Buscaremos

Poder mensurar que nos diga que tan lejos esta B de estar ordenada en forma ascendente.

Si

$b_i < b_{i+1}$ para todo i , entonces A y B son iguales

Queremos que el valor de “diferencia” sea igual a cero

A medida

Que B esté “más mezclado” el valor de diferencia debe aumentar

Inversiones

Utilizaremos

El concepto de “inversiones” para medir cuando desordenado esta la lista B

Dos elementos b_i, b_j con $i < j$ están invertidos

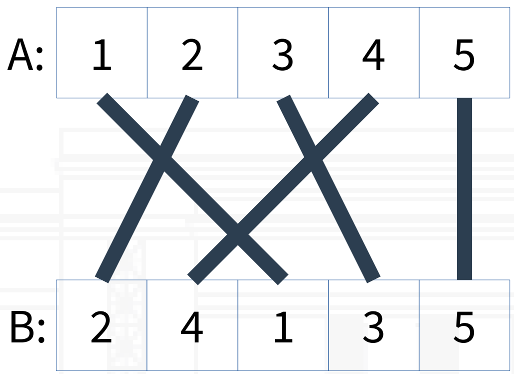
Si $b_i > b_j$

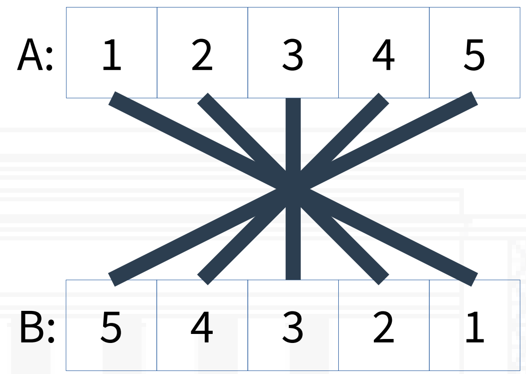
Podemos – mediante fuerza bruta – calcular la cantidad de inversiones

Comparando cada posición con todas las siguientes → $O(n^2)$

¿Podemos hacerlo mejor?

Ejemplo





Podemos ver que

1 está "invertido" con 2 elementos: 2 y 4 → (2,1) (4,1)

4 está "invertido" con 1 elemento: 3 → (4,3)

En total hay 3 inversiones

1 está invertido con 4 elementos

2 está invertido con 3 elementos

3 está invertido con 2 elementos

4 está invertido con 1 elementos

En total hay 10 inversiones $\binom{n}{2}$

Idea de la solución

Si la lista de preferencias esta semi-ordenada en mitades

Con ausencia de inversiones dentro de las mitades de la lista

Las inversiones se encuentran

entre elementos de la primera mitad con elementos de la segunda.

Podemos contar las inversiones realizando un proceso de merge

Contando - cada vez que se inserta un elemento de la segunda mitad - cuantos elementos de la primera mitad quedan por insertarse

Este proceso se puede realizar en O(n)

Idea de la solución (cont.)

El resultado del proceso de merge y conteo

- Sera una lista ordenada

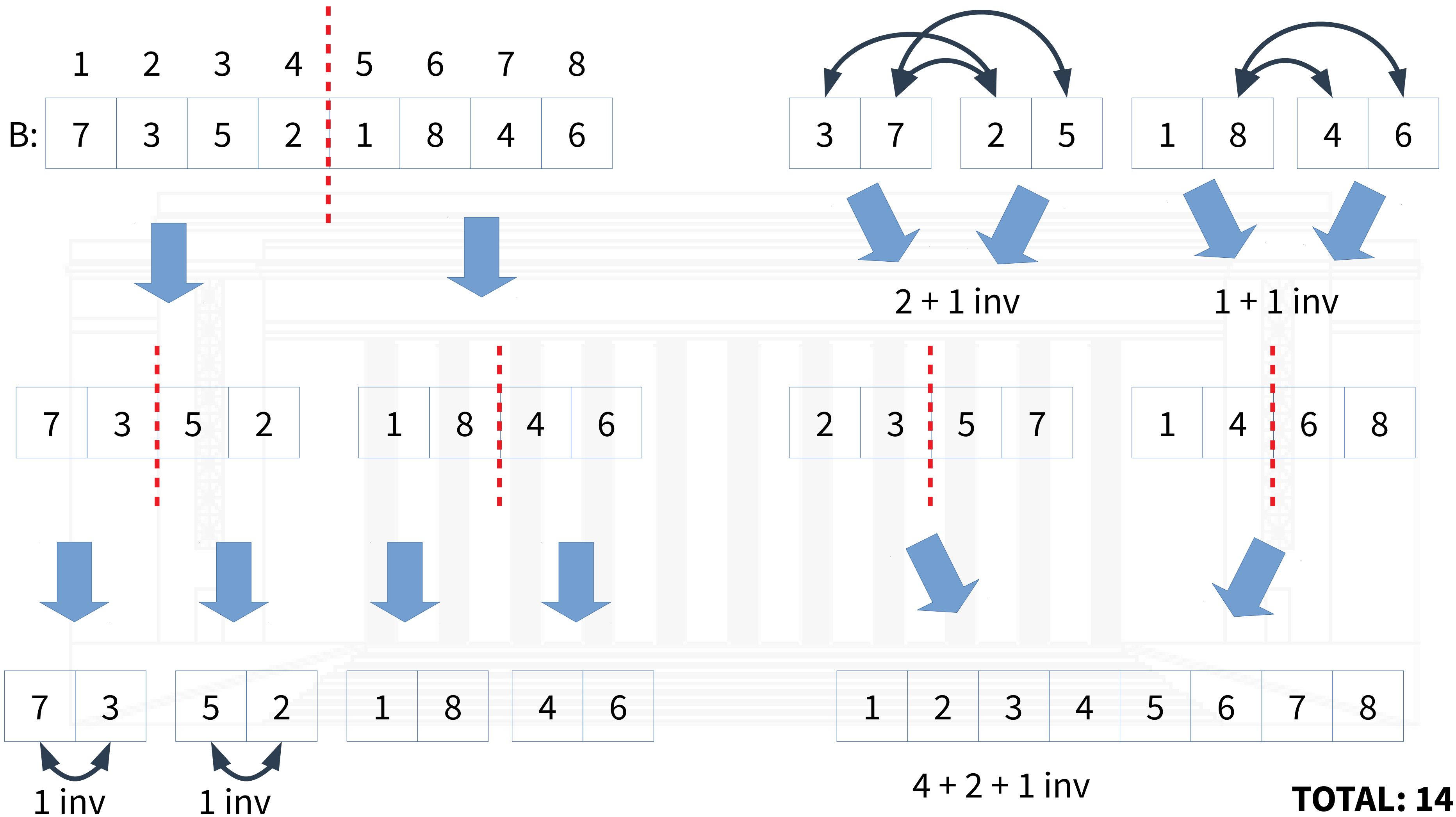
Lamentablemente no podemos suponer

- Que las mitades estén ordeandas

Pero, podemos partir estas partes recursivamente para hacerlo.

- La suma de todas las inversiones contadas en cada uno de los subproblemas corresponden a la cantidad total de inversiones en la lista

Ejemplo



1 2 3 4 5 6 7 8
B: 7 3 5 2 1 8 4 6
3 7 2 5
1 8 4 6
2 + 1 inv
1 + 1 inv
7 3 5 2
1 8 4 6
2 3 5 7
1 4 6 8
7 3
5 2
1 8
4 6
1 2 3 4 5 6 7 8
1 inv
1 inv
4 + 2 + 1 inv
TOTAL: 14

Pseudocódigo

```
Ordenar-Contar(L)
    Si |L|=1
        Retornar (0,L)  // No hay inversiones

    Sino
        Sea A los techo(n/2) primeros elementos de L
        Sea B los piso(n/2) restantes elementos de L

        (ra,A) = Ordenar-Contar(A)
        (rb,B) = Ordenar-Contar(B)

        (r,L)  = Merge-Contar(A,B)

    Retornar (r+ra+rb,L)
```

Pseudocódigo (cont.)

```
Merge-Contar(A,B)
    Sea L lista
    inv =0
    j=0, i=0        //punteros a la lista A y B
    Repetir
        a = A[i],  b = B[j]
        Si a>b
            L[i+j]=a , i++
        Sino
            L[i+j]=b , j++
            inv=+ (|A|- i)
    Mientras i<|A| y j<|B|
    Desde i hasta |A|-1
        L[i+j]=A[i]
        inv+=|B|
    Desde j hasta |B|-1
        L[i+j]=B[i]
    Retornar (inv,L)
```

Complejidad

Cada problema con n elementos

- Se divide en 2 subproblemas de n/2 elementos

La unión de los resultados

- Se construye recorriendo unas vez los n elementos → $O(n)$

Podemos expresar la recurrencia como

- $T(n) = 2T(n/2) + O(n)$

Utilizando el teorema maestro

- Nos queda una complejidad temporal $T(n) = O(n \log n)$

Presentación realizada en Septiembre de 2020