

Algoritmos Divide y vencerás

Comparación de preferencias

- Elvira Castillo Fernández
- David Gil Bautista
- José Luis Izquierdo Mañas
- Freddy A. Jaramillo López
- Alejandro Jerónimo Fuentes
- Gregorio Vidoy Fajardo



UNIVERSIDAD
DE GRANADA

Comparación de preferencias

- Problema
- Solución fuerza bruta
- Solución Divide y Vencerás
- Eficiencia fuerza bruta
- Eficiencia Divide y Vencerás

Comparación de preferencias

Descripción del problema

Una web quiere comparar las preferencias de un usuario con la de otros:

- Se establece un ranking de n productos.
- La app consulta su BD para encontrar usuarios con gustos similares.

Tenemos que medir el número de inversiones entre dos rankings:

- Ranking A: $1, 2, 3, \dots, n$.
- Ranking B: $a_1, a_2, a_3, \dots, a_n$.
- i y j están invertidos si $i < j$ pero $a_i > a_j$

Comparación de preferencias

Solución fuerza bruta

	P1	P2	P3	P4	P5	P6	P7	P8	P9
A	0	1	2	3	4	5	6	7	8
B	0	1	3	4	2	5	6	7	8



Inversiones

3-2 4-2

Algoritmo por fuerza bruta:

Comprobar todos los pares (i,j)

$O(n^2)$

Comparación de preferencias

Solución Divide y vencerás

Dividimos la lista del ranking en dos mitades y contamos de forma recursiva el numero de inversiones en cada mitad.

Contamos las inversiones en las que a_i y a_j están en mitades diferentes y devolvemos la suma de las cantidades

Comparación de preferencias

Solución Divide y vencerás

0	1	2	3	4	5	6	7	8	9
0	5	4	1	2	3	6	7	8	9

Dividimos el problema : $2T(n/2)$

0	5	4	1	2	3	6	7	8	9
---	---	---	---	---	---	---	---	---	---

5 inversiones

0 inversiones

5-4 5-1 5-2 4-1 4-2

Total: $5+0+7$

Comparación de preferencias

Solución Divide y vencerás

Ordenamos cada mitad y contabilizamos las inversiones en las que a_i y a_j están en mitades diferentes y a continuación se mezclan las dos mitades para devolver un conjunto ordenado

0	1	2	4	5	3	6	7	8	9
					2	0	0	0	0

$$T(n) \leq T(n/2) + T(n/2) + O(n) \quad \longrightarrow \quad T(n) = O(n \log n)$$

Comparación de preferencias

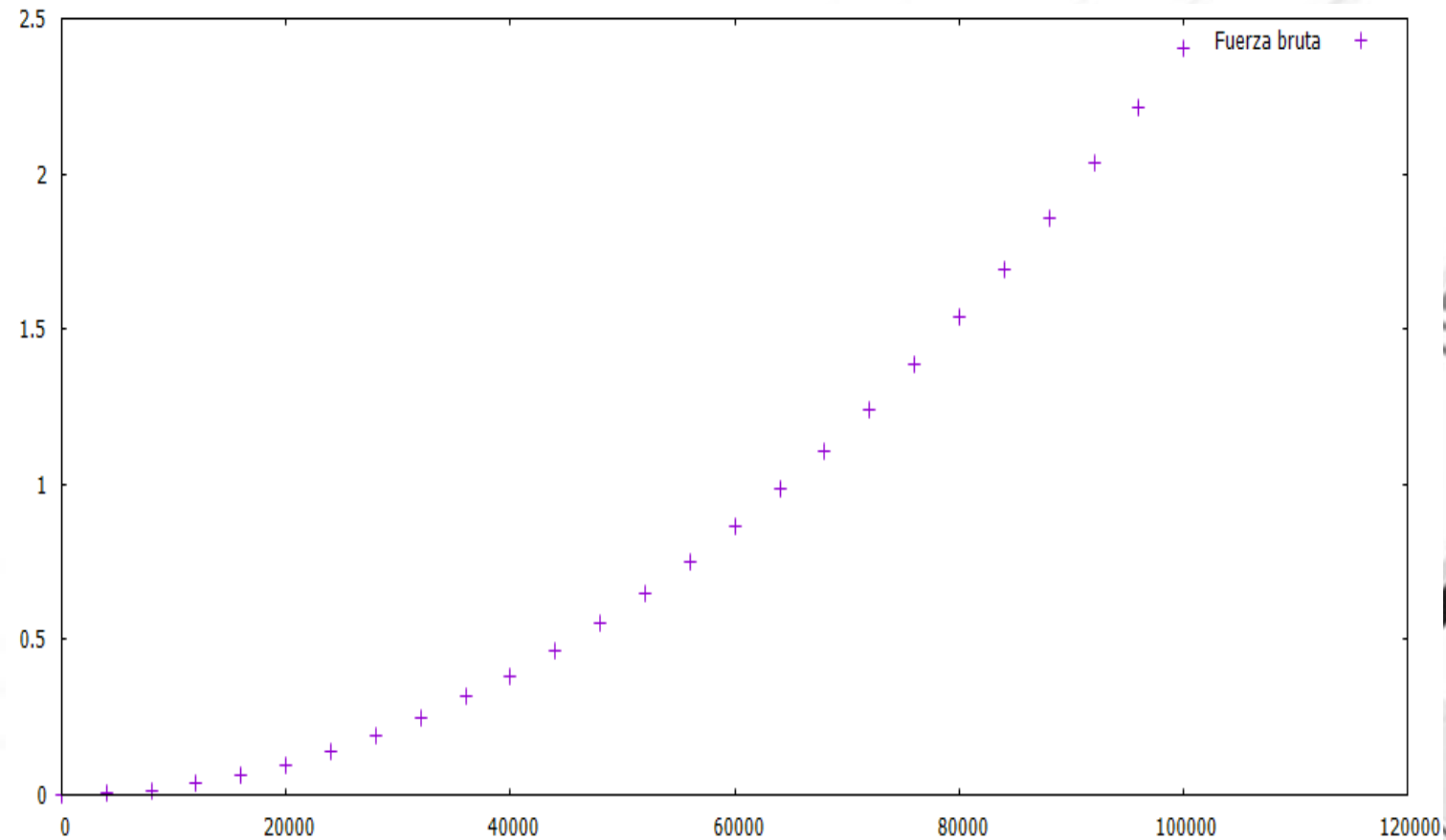
Solución Divide y vencerás

Implementación:

```
DyVPreferencias(V){  
    si (v.lenght==1)  
        return (0,V);  
    Divide en dos mitades el vector A y B;  
    (resula,A) <- DyVPreferencias(A);  
    (resulb,B) <- DyVPreferencias(B);  
    (resul,V)<-OrdenayCuentaInversiones(A,B);  
  
    return (ra+rb+r, V);  
}
```

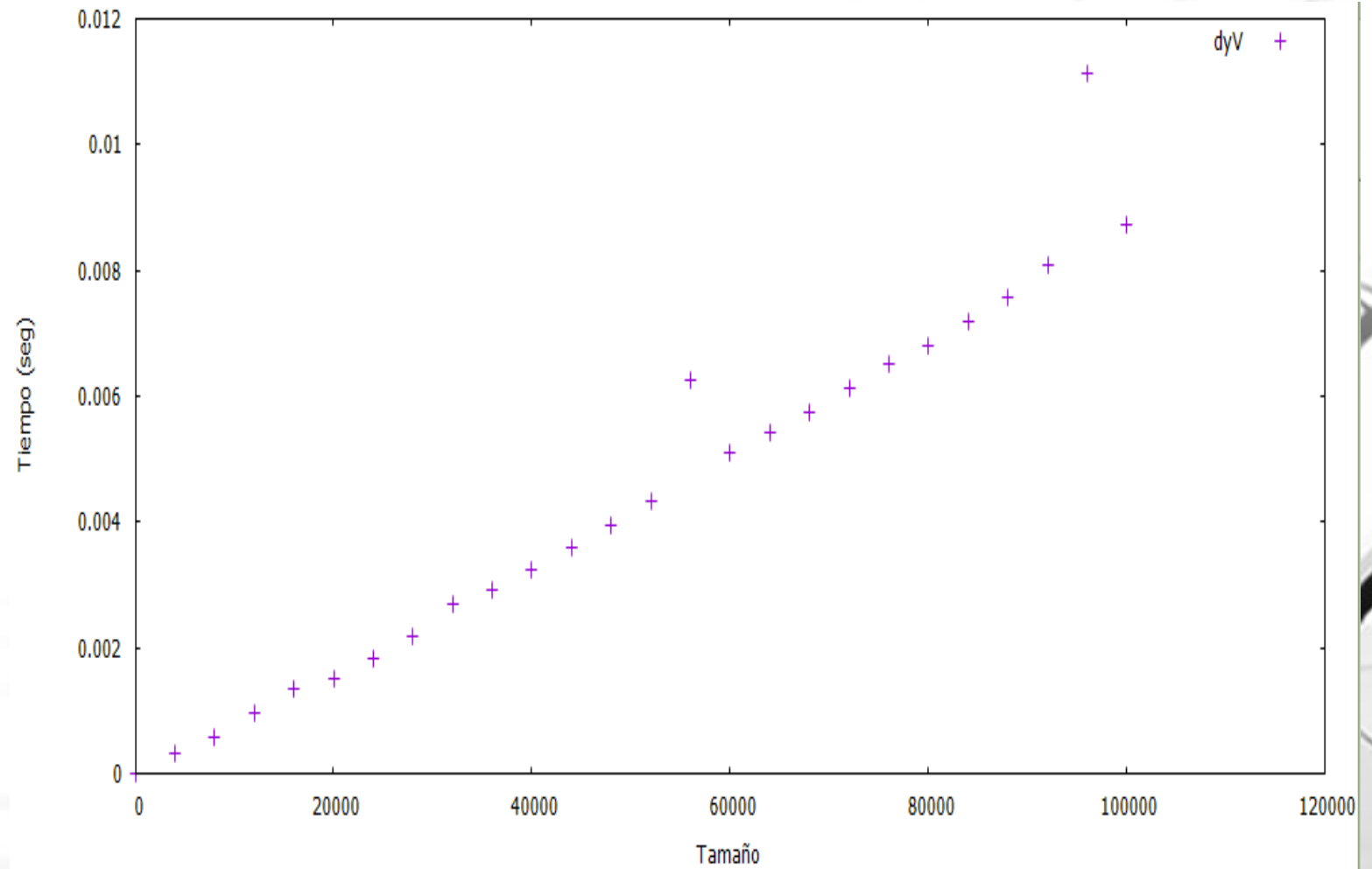

Eficiencia fuerza bruta

- El algoritmo posee una eficiencia de $O(n^2)$
- Está compilado con optimización -O2
- Los valores van desde 1 hasta 100000 con un incremento de 4000



Eficiencia Divide y vencerás

- El algoritmo posee una eficiencia de $O(n \log n)$
- Está compilado con optimización -O2
- Los valores van desde 1 hasta 100000 con un incremento de 4000

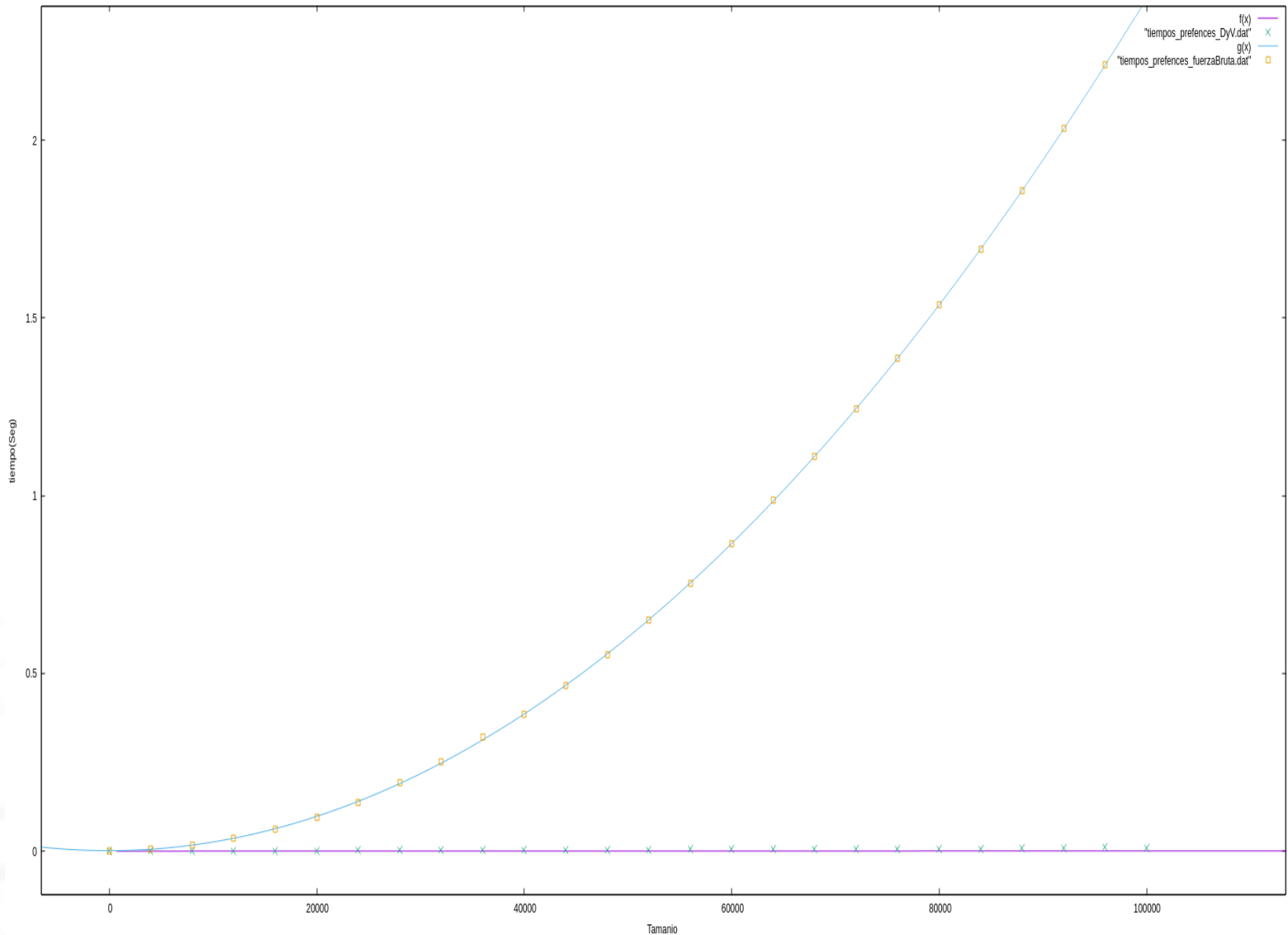


Eficiencia híbrida de ambos algoritmos

- En esta gráfica mostramos como se ajustan las funciones
- $F(x) = a_0 \cdot x^2 + a_1 \cdot x + a_2$
- $G(x) = a \cdot x \cdot (\log_{10}(b \cdot x) / \log_{10}(2)) + c$

La función del fuerza bruta se ajusta perfectamente a $O(n^2)$

La función del algoritmo dyV se ajusta muy bien a $O(n \log n)$



Eficiencia híbrida: fuerza bruta

$O(n^2)$

Calculamos los coeficientes para el ajuste a

- $F(x) = a_0 \cdot x^2 + a_1 \cdot x + a_2$

iter	chisq	delta/lim	lambda	a	b	d
0	1.4145008888e-04	0.00e+00	6.37e-01	2.395163e-10	3.778314e-08	1.170323e-03
1	1.4145008888e-04	-8.43e-10	6.37e-02	2.395163e-10	3.778314e-08	1.170323e-03

After 1 iterations the fit converged.

final sum of squares of residuals : 0.00014145

rel. change during last iteration : -8.4314e-15

degrees of freedom (FIT_NDF) : 23
rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.00247992
variance of residuals (reduced chisquare) = WSSR/ndf : 6.15e-06

Final set of parameters		Asymptotic Standard Error	
=====		=====	
a	= 2.39516e-10	+/- 6.055e-13	(0.2528%)
b	= 3.77831e-08	+/- 6.269e-08	(165.9%)
d	= 0.00117032	+/- 0.001354	(115.7%)

correlation matrix of the fit parameters:

	a	b	d
a	1.000		
b	-0.966	1.000	
d	0.716	-0.846	1.000

Eficiencia híbrida dyV $O(n \log n)$

Calculamos los coeficientes para el ajuste a

$$F(x) = a \cdot x \cdot (\log_{10}(b \cdot x) / \log_{10}(2)) + c$$

iter	chisq	delta/lim	lambda	a	b
0	8.7988883257e-06	0.00e+00	3.72e-03	5.576237e-09	-4.800191e-05
1	8.7988883257e-06	-7.70e-11	3.72e-04	5.576237e-09	-4.800191e-05

After 1 iterations the fit converged.

final sum of squares of residuals : 8.79889e-06

rel. change during last iteration : -7.70127e-16

degrees of freedom (FIT_NDF) : 24

rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.000605492

variance of residuals (reduced chisquare) = WSSR/ndf : 3.6662e-07

Final set of parameters		Asymptotic Standard Error	
=====		=====	
a	= 5.57624e-09	+/- 2.352e-10	(4.218%)
b	= -4.80019e-05	+/- 0.0002217	(461.9%)

correlation matrix of the fit parameters:

	a	b
a	1.000	
b	-0.844	1.000

FIN

Gracias por su atención