Práctica 2:
Algoritmos divide y vencerás

Grupo Ciri

MARTÍN QUIRÓS, JUAN ANTONIO
MARTÍNEZ IÁÑEZ, GONZALO
POZO RAMÍREZ, RAFAEL
SANTOS SALVADOR, JOSÉ
SORIA GONZÁLEZ, RAÚL

Objetivos

- Convertir un algoritmo en divide y vencerás.
- Calcular la matriz traspuesta en c++ mediante un algoritmo de divide y vencerás.
- Realizar un algoritmo de divide y vencerás para el problema de Comparación de preferencias.

Matriz traspuesta

Eficiencia teórica

Código de la traspuesta sin Divide y vencerás.

```
for(int i = 0; i < n; i++)
   for(int j = 0; j < n; j++)
      matriz_traspuesta[j][i] = matriz_original[i][j];
```

La eficiencia es $O(n^2)$.

Matriz traspuesta

Código de la traspuesta con Divide y Vencerás.

```
void traspuesta(int **m, int **res, int inif, int finf, int inic, int finc){
   if(inif == finf)
        res[inif][inic] = m[inic][inif];
   else{
        int centrof = (inif + finf)/2;
        int centroc = (inic + finc)/2;

        traspuesta(m, res, inif, centrof, inic, centroc);
        traspuesta(m, res, inif, centrof, centroc + 1, finc);
        traspuesta(m, res, centrof + 1, finf, inic, centroc);
        traspuesta(m, res, centrof + 1, finf, centroc + 1, finc);
   }
}
```

Matriz traspuesta

$T(n) = 4T(n/4) + cn$

Cambio de variable, $n = 4^k$
$T(4^k) = 4T(4^k/4) + c4^k$; $T(4^k) = 4T(4^{k-1}) + c4^k$

Cambio de variable, $T(4^k) = t_k$
$t_k = 4t_{k-1} + c4^k$; $t_k - 4t_{k-1} = c4^k$

Ecuación característica: $(x - 4)(x - 4) = 0$; $(x - 4)^2 = 0$
$t_k = c_1 4^k + c_2 k 4^k$

Deshacer el cambio $k = \log_4 n$
$t_k = c_1 4^{\log_4 n} + c_2 (\log_4 n) 4^{\log_4 n}$
$t_k = c_1 n^{\log_4 4} + c_2 \log_4 n (n^{\log_4 4})$
$t_k = c_2 n(\log_4 n) + c_1 n$

La eficiencia es $O(n*\log_4 n)$

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Resultado

0	8						
1	9						

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Resultado

0	8	16	24				
1	9	17	25				

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Resultado

0	8	16	24				
1	9	17	25				
2	10						
3	11						

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Resultado

0	8	16	24				
1	9	17	25				
2	10	18	26				
3	11	19	27				

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Resultado

0	8	16	24				
1	9	17	25				
2	10	18	26				
3	11	19	27				
4	12	20	28				
5	13	21	29				
6	14	22	30				
7	15	23	31				

Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Resultado

0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28				
5	13	21	29				
6	14	22	30				
7	15	23	31				

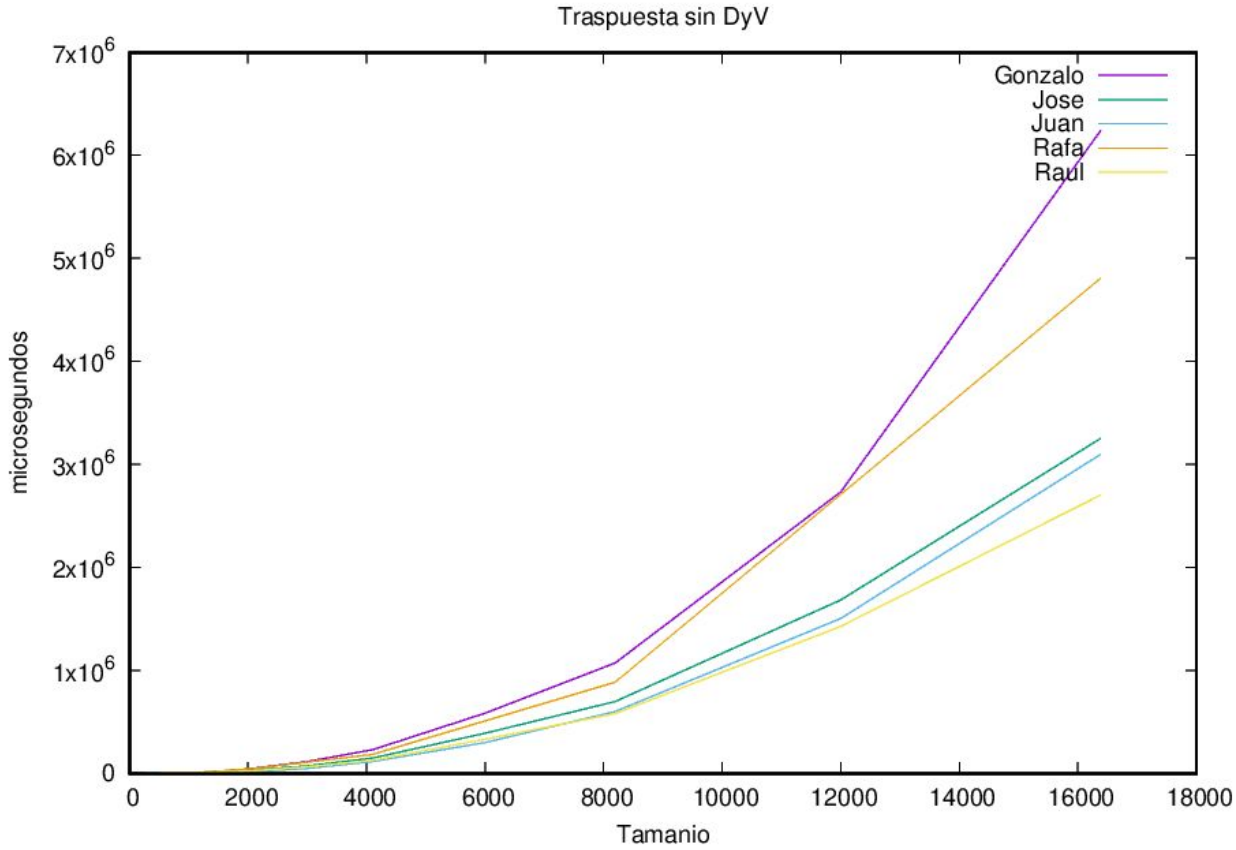
Original

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

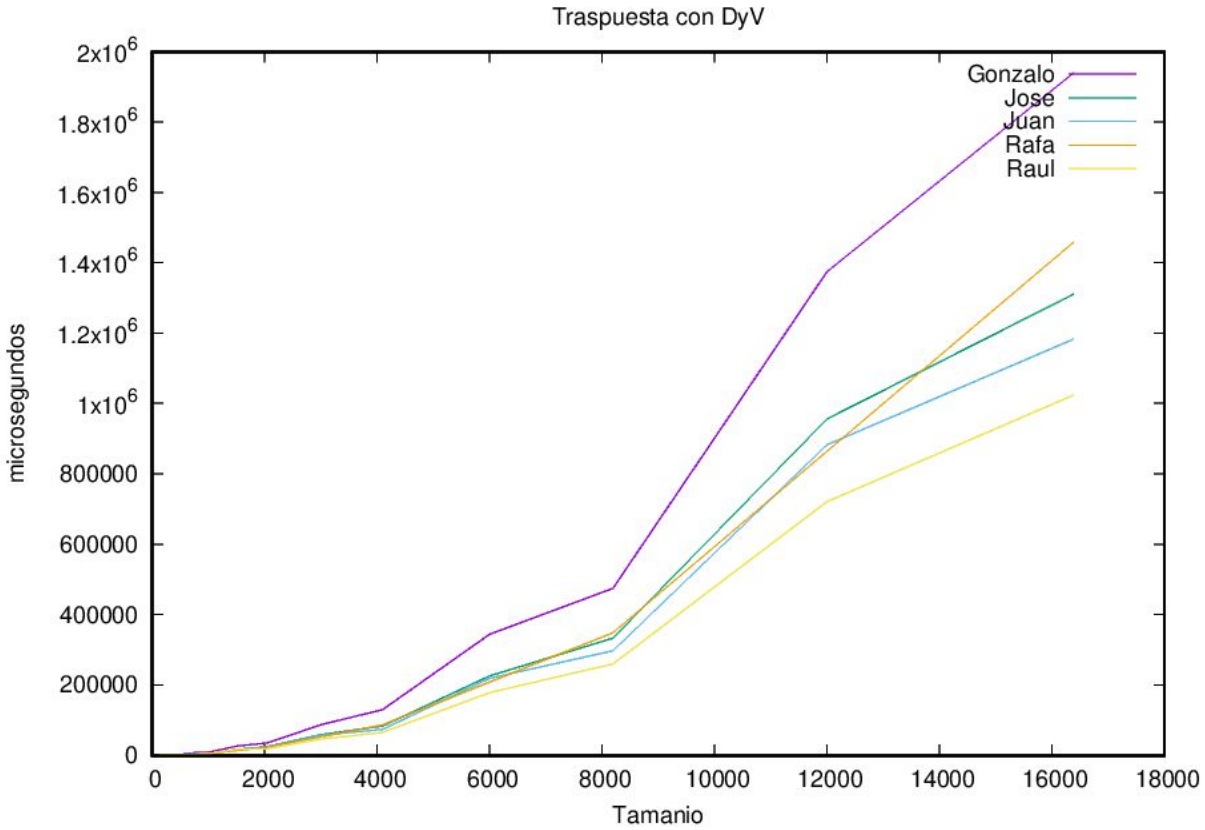
Resultado

0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

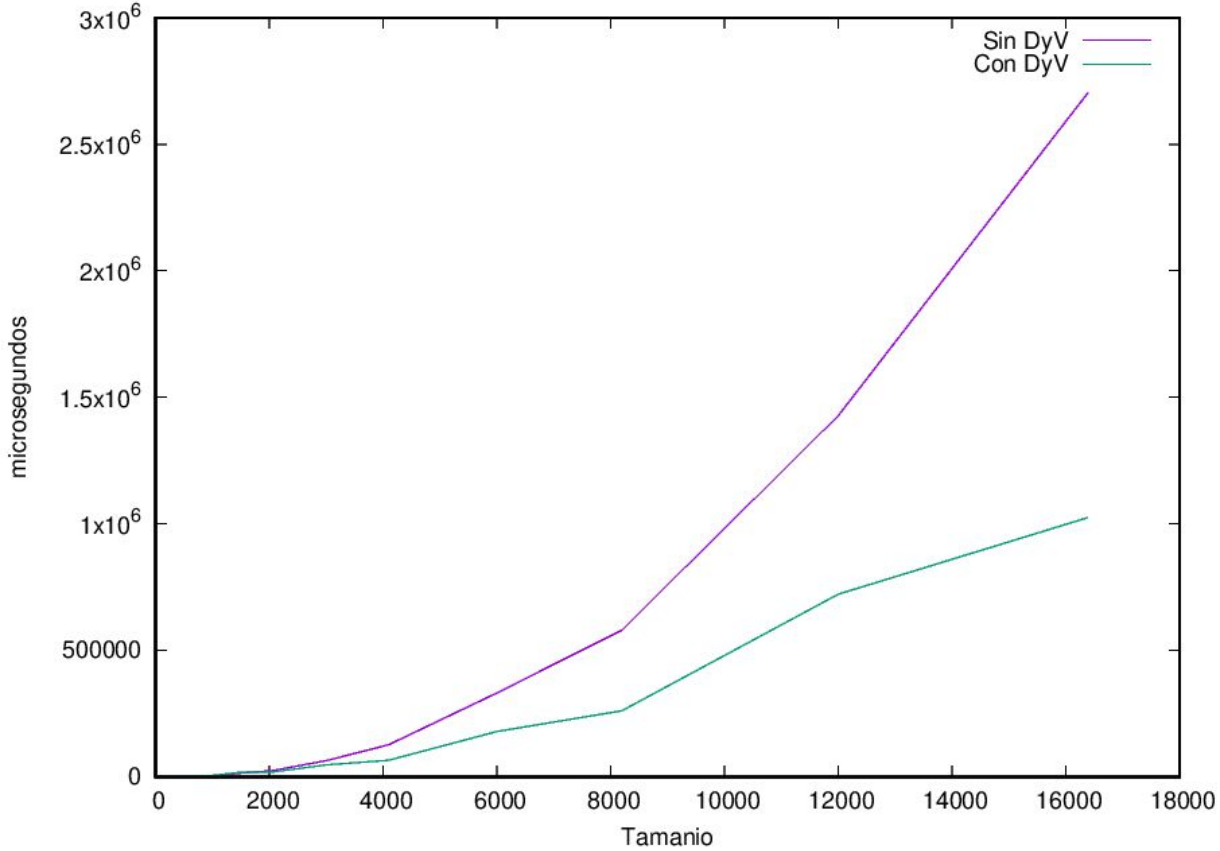
Eficiencia empírica



Eficiencia empírica



Eficiencia empírica

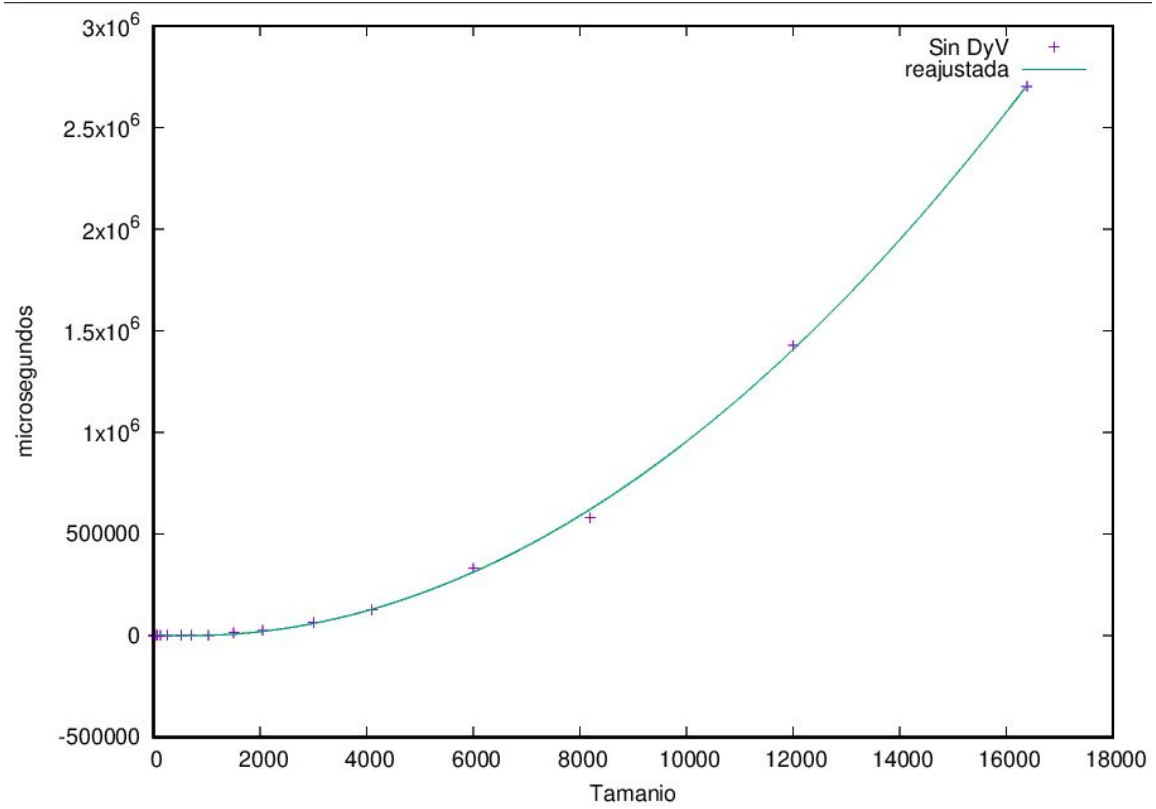


	Sin DyV	Con DyV
512	321	1271
1024	1898	5285
2048	23876	18526
4096	126280	65017

Eficiencia híbrida

$f(x) = a_0*x^2 + a_1*x + a_2$

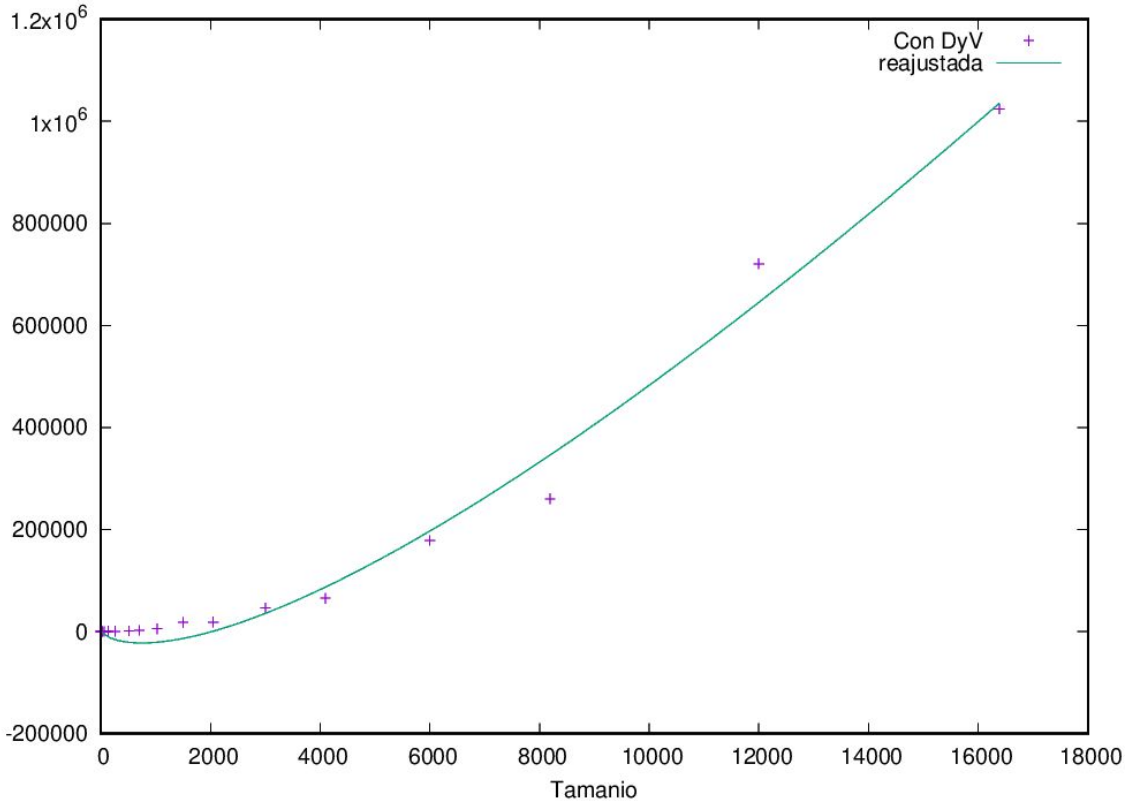
a0 = 0.0109336
a1 = -14.1681
a2 = 2691.4



Eficiencia híbrida

$f(x) = a_0 * x * \log_4 x + a_1 * x$

$a_0 = 41.9085$
$a_1 = -230.173$



Comparación de preferencias

Planteamiento del problema

El algoritmo de comparación de preferencias se basa en calcular qué tan parecidos son las “preferencias” de 2 usuarios contando el número de inversiones que tienen entre sí.

Para realizar ésto, se realiza calculando si dos preferencias están o no invertidas entre sí. Por ej: El usuario A, “prefiere” a ntes el valor 1 al 4 y el usuario B, prefiere antes el valor 4 al 1, así pues tendrían preferencias invertidas y viceversa.

A	B
1	3
2	1
3	5
4	2
5	4

Cambio
Próximo cambio

Caso inicial

A	B
1	3
2	1
3	5
4	2
5	4

A	B
1	3
2	1
3	5
4	2
5	4

A	B
1	1
2	3
3	5
4	2
5	4

+1

A	B
1	1
2	2
3	3
4	5
5	4

+2

A	B
1	1
2	2
3	3
4	4
5	5

+1

Eficiencia teórica

$T(n) = 2T(n/2) + cn$

Cambio de variable, $n = 2^k$.
$T(2^k) = 2T(2^k/2) + c2^k$; $T(2^k) = 2T(2^{k-1}) + c2^k$

Cambio de variable, $T(2^k) = t_k$.
$t_k = t_{k-1} + c2k$; $t_k - t_{k-1} = c2^k$

Ecuación característica: $(x - 2)^2 = 0$.
$t_k = c_1 2^k + c_2 k 2^k$

Deshacer el cambio de variable, $k = \log_2 n$
$t_k = c_1 2^{\log 2n} + c_2 \log_2 n(2^{\log 2n})$
$t_k = c_1 n + c_2 n(\log_2 n)$

La eficiencia es $O(n*\log_2 n)$

FIN