

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo
DyV

Comparación

Práctica 2: Divide y vencerás

Algorítmica

Universidad de Granada

5 de abril de 2016

Índice

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo DyV

Comparación

1 Introducción

2 Ejercicio

3 Fuerza bruta

4 Algoritmo DyV

5 Comparación

Introducción

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo
DyV

Comparación

- El objetivo de ésta práctica era resolver uno de los cinco problemas dados aplicando la técnica divide y vencerás.
- Además de la elaboración del programa, hemos calculado las distintas eficiencias, comparándolo también con un algoritmo de ordenación por fuerza bruta.
- Para medir tiempos hemos utilizado la biblioteca de C++ más moderna y precisa destinado a obtener tiempos de reloj: la biblioteca **chrono**

Enunciado del ejercicio

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo
DyV

Comparación

Dados K vectores de longitud N ordenado cada uno de ellos obtener un tamaño de $N \cdot K$ ordenado.

Pseudocódigo

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo
DyV

Comparación

Algoritmo Fuerza bruta.

Require: Vectores ordenados, numero de estos y tamaño

$V1 = V[0]$

$Vfinal = V[1]$

$Vfinal = \text{OrdenarVectores}(V1, Vfinal)$

for $i=2$ hasta $i=k-1$ **do**

$V1=V[i];$

$Vfinal= \text{OrdenarVectores}(V1, Vfinal);$

$i++;$

end for

Eficiencia

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo DyV

Comparación

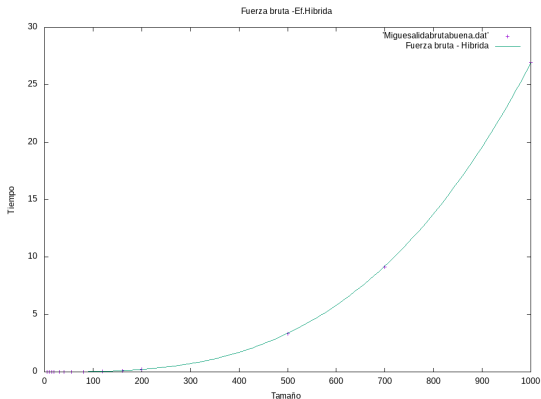


Figura: La eficiencia resultante es de n^3 cubo

Pseudocódigo

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo
DyV

Comparación

La estructura y el funcionamiento de nuestro código es el siguiente:

Recursivo(matriz)

Require: Matriz de vectores

if Si el número de vectores menor o igual a 1 **then**

return La matriz con una fila

else Si el número de vectores es mayor que 1

middle = n° filas / 2

Up = matriz[:middle][num_colum]

Down = matriz[middle:][num_colum]

Up = Recursivo(Up)

Down = Recursivo(Down)

Result = Merge(Up, Down)

return Result

end if

Eficiencia

Practica 1

Algorítmica

Algoritmo
DyV

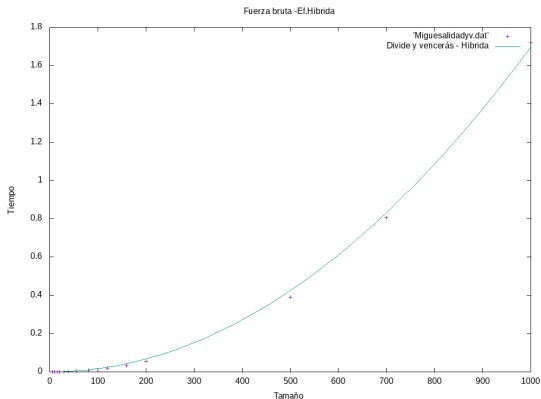


Figura: La eficiencia obtenida ha resultado ser de n cuadrado, mejorando en un orden al de fuerza bruta.

Comparación

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo
DyV

Comparación

En esta última sección, hemos comparado los dos algoritmos que hemos desarrollado, veremos si efectivamente o no nuestra implementación utilizando un enfoque divide y vencerás obtenemos mejoras con respecto a uno de fuerza bruta.

Comparación

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo DyV

Comparación

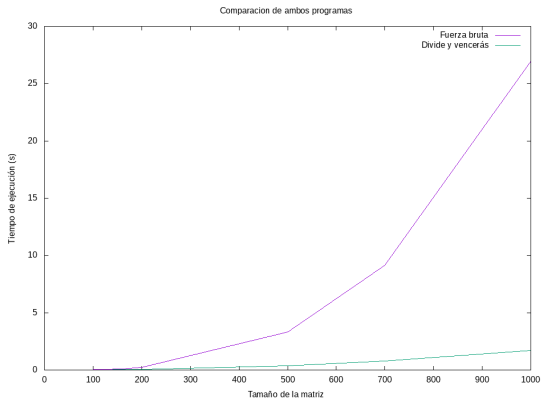


Figura: Figura perteneciente a la comparación entre ambas eficiencias

Porcentaje de error y constantes ocultas

Practica 1

Algorítmica

Introducción

Ejercicio

Fuerza bruta

Algoritmo DyV

Comparación

Algoritmo	Constante Oculta	Error
DyV	$a_0 = 1.69804e-06$	$\pm 1.226e-08$ (0.7218 %)
Fuerza bruta	$a_0 = 2.69052e-08$	$\pm 2.352e-11$ (0.0874 %)