Práctica 2 El elemento en su posición

María Jesús López Salmerón Nazaret Román Guerrero Laura Hernández Muñoz José Baena Cobos Carlos Sánchez Páez

6 de abril de 2018

Presentación del problema Diseño de algoritmos Algoritmo clásico Algoritmo Divide y Vencerás Medición de tiempos Estudio de eficiencia Eficiencia empírica

Eficiencia empírica

Tamaños de problema
Comparación entre
algoritmos
Calculo de la eficiencia
nibrida

Errores en el cálculo d
constante oculta

constante oculta
Vectores con elementos
repetidos
Vectores con elementos
repetidos

- Presentación del problema
- 2 Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- 3 Estudio de eficiencia
 - Eficiencia empírica
 - Tamaños de problema
 - Ocomparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos
 - Vectores con elementos repetidos

El elemento en su posición

```
Presentación del problema
Diseño de algoritmos
Algoritmo cisicio
Algoritmo Divide y Vencer
Medición de tiempos
Extudio de eficiencia
Eficiencia empirica
Tamaños de problema algoritmos entre
algoritmos
Cálculo de la eficiencia
Emores en el cálculo de la
Constante contra
```

Sea v un vector **ordenado** y sin elementos repetidos, determinar si $\exists i : v[i] = i$

Ejemplos

Presentación del problema

0 5 6 8 9 -70|-35| 3 15 25 51 68 75 83

Tamaño del vector: 10

Resultado del algoritmo : 3

Tamaño del vector: 10

Resultado del algoritmo : -1

Diseño de algoritmos

Algoritmo Clásico
Algoritmo Divide y Ven
Medición de tiempos

Estudio de eficiencia

Eficiencia empirica

Zamaños de problema
Lomparación entre
algoritmos

Cálculo de la eficiencia
hibrida

Errores en el cálculo constante oculta l'ectores con elementos epetidos Vectores con elemento repetidos

- Presentación del problema
- 2 Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- Studio de eficiencia
 - Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos
 - Vectores con elementos repetidos

Clásico

```
Algoritmo Clasco
Algoritmo Divide y Venci
Medición de tiempos

istudio de eficiencia

Eficiencia empírica

Tamaños de problema
companion entre
Calculo de la eficiencia
hibrida

Errores en el calculo de
constante coult

fectores con elementos
epetidos

Vectores con elementos

vectores con elementos
```

Presentación del problema Diseño de algoritmos Algoritmo clásico

Algoritmo Divide y Vence Medición de tiempos studio de eficiencia

Eficiencia empírica

Tamaños de problem
Comparación entre
algoritmos
Cálculo de la eficiencia
hibrida

Errores en el cálculo

Vectores con elemento repetidos Vectores con element repetidos Presentación del problema

- 2 Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- 3 Estudio de eficiencia
 - Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos
 - Vectores con elementos repetidos

Divide y Vencerás

```
Presentación del problema
Diseño de algoritmos
Algoritmo cislos
Algoritmo cislos
Algoritmo Divide y Vencerás
Medición de tiempos
Estudio de el eficiencia
Eficiencia empirica
Finando de problema
algoritmos entre
Calcido de la eficiencia
ber presente esta de la eficiencia
contante codia
reference nel ejiculio de la
contante codia
reference nel estementos
Ventores con elementos
```

```
int elementoEnSuPosicion(const vector<int> &v, const int ini,
                const int fin) {
  if (ini == fin) { //Caso base
  if (v[ini] == ini)
    return ini:
 else
   return -1;
  else { //Buscamos en la mitad adecuada.
    int mitad = (ini + fin) / 2;
    if (v[mitad] == mitad)
     return mitad;
    else if (v[mitad] > mitad)
     return elementoEnSuPosicion(v, ini, mitad-1);
    else
     return elementoEnSuPosicion(v, mitad + 1, fin);
```

Presentación del problema

Diseño de algoritmos

Algoritmo clásico

Algoritmo Divide y Vencer.

Medición de tiempos

Estudio de eficiencia

Tamaños de problem Comparación entre algoritmos Cálculo de la eficienci híbrida Errores en el cálculo constante oculta

/ectores con elementos epetidos Vectores con elemento repetidos

- Presentación del problema
- Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- Bestudio de eficiencia
 - Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 Cálculo de la eficiencia híbrida
 - Calculo de la efficiencia hibrida
 Errores en el cálculo de la constante oculta
 - Enores en el calcalo de la constante ocure
- 4 Vectores con elementos repetidos
 - Vectores con elementos repetidos

Modificación de código fuente

issello de algoritmos
Migoritmo Cidsico
Migoritmo Divide y Vencerás
Medición de tiempos
studio de eficiencia
studio de eficiencia
supplica
Tannalios de problema
suportiros
algoritmos
suportiros
Errores en el cálculo de la
contrate do culto
contrate do culto
problema
ventra de problema
suportiros
ventras en el cálculo de la
contrate do culto
problema
ventras en el cálculo de la
contrate con elementos
protidos
ventroses con elementos
protidos

```
high_resolution_clock::time_point tantes;
high_resolution_clock::time_point tdespues;
duration < double > tiempo;
double acumulado = 0;
int pos;
for (int i = 0; i < 1000; i++) {
       tantes = high_resolution_clock::now();
       pos = elementoEnSuPosicion(myvector);
       tdespues = high_resolution_clock::now();
       tiempo = duration_cast<duration<double>>
                        (tdespues - tantes);
       acumulado += tiempo.count();
acumulado /= 1000;
```

Presentación del problema

Diseño de algoritmos

Algoritmo clásico

Algoritmo Divide y Vencerá:

Medición de tiempos

Estudio de aficiancia

Tamaños de probler Comparación entre algoritmos Cálculo de la eficienc híbrida Errores en el cálculo constante oculta

Vectores con elemento repetidos Vectores con element repetidos Presentación del problema

- Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- 3 Estudio de eficiencia
 - Eficiencia empírica
 Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos
 - Vectores con elementos repetidos

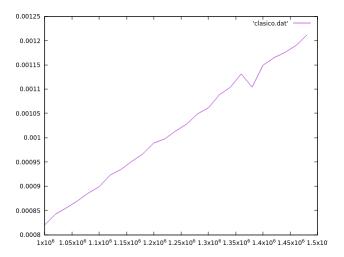
Tamaños de problema

Addición de tiempos tudio de eficiencia ficiencia empirica Tamalos de problema Comparación entre algoritmos Siducio de la eficiencia bibida Errores en el cálculo de constante cuita cortes con el ementos

Vectores con elementos repetidos Vectores con elementos repetidos Fin de la presentación

Algoritmo	Eficiencia	Tamaño inicial	Tamaño final	Incremento
Clásico	O(n)	1.000.000	1.480.000	20.000
Divide y Vencerás	O(log(n))	1.000.000	13.000.000	500.000

Algoritmo clásico



Algoritmo Divide y Vencerás

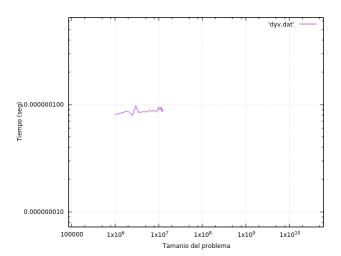
Presentación del problema
Diseño de algoritmos
Algoritmo Clásico
Algoritmo Divide y Vencerás
Medición de tiempos
Estudio de eficiencia
Eficiencia empírica

Tamaños de problem Comparación entre algoritmos Cálculo de la eficiencia híbrida

Errores en el cálculo constante oculta

repetidos Vectores con elemen

repetidos Fin de la presentación



Algoritmo Divide y Vencerás (zoom)

Presentación del problema Diseño de algoritmos Algoritmo clásico Algoritmo Divide y Vencerás Medición de tiempos Estudio de eficiencia

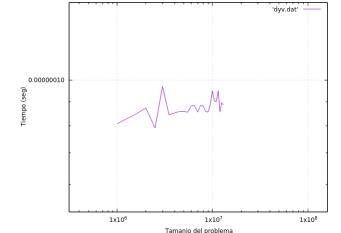
Eficiencia empirica

Tamaños de problem
Comparación entre
algoritmos
Cálculo de la eficiencia
hibrida

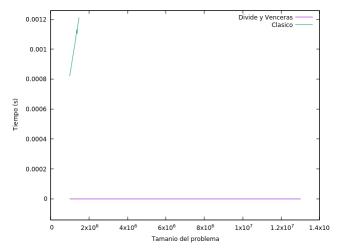
hibrida Errores en el cálculo constante oculta

repetidos

Vectores con elemen repetidos



Comparación entre ambos algoritmos



Presentación del problema

Diseño de algoritmos

Algoritmo clásico

Algoritmo Divide y Vencerá

Medición de tiempos

Estudio de eficiencia

Tamaños de problema Comparación entre algoritmos Cálculo de la eficiencia híbrida Errores en el cálculo d constante oculta

Vectores con elementos repetidos Vectores con elemento repetidos Presentación del problema

- Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- 3 Estudio de eficiencia
 - Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos
 - Vectores con elementos repetidos

Errores en el cálculo de la constante oculta

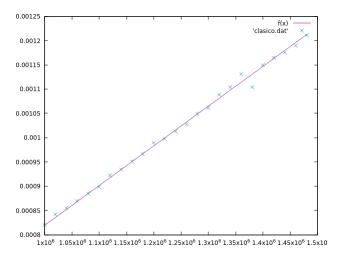
tudio de eficiencia Eficiencia empírica Tamaños de problema Compazación entre algoritmos algoritmos algoritmos Enrores en el cálculo de la constante oculta stores con elementos petidos fectores con elementos petidos

Algoritmo	Eficiencia teórica	Valor de la constante oculta	Error
Clásico	O(n)	8.19304 ·10 ⁻¹⁰	0.1441 %
Divide y Vencerás	O(log(n))	$5.61125 \cdot 10^{-9}$	0.9174%

Algoritmo clásico

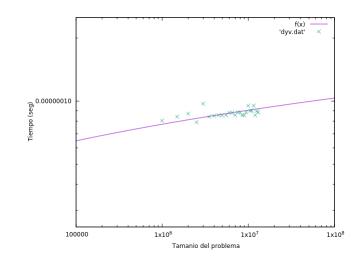
```
esentación del problema
seño de algoritmos
Algoritmo Clásico
Algoritmo Divide y Vencera
Medición de tiempos
studio de eficiencia
Eficiencia empírica
Tamaños de problema
algoritmos
Eficiencia de la eficiencia
bibrida
```

Vectores con elementos repetidos Vectores con elementos repetidos



Algoritmo Divide y Vencerás

Presentación del problema
Diseño de algoritmos
Algoritmo cislacio
Algoritmo cislacio
Algoritmo cislacio
Algoritmo cislacio
Algoritmo Divide y Vencerás
Medición de tiempos
Eficiencia empirica
Eficiencia empirica
Tamalias de problema
Complexación entre
Calicalo de la eficiencia
Britista
Errorese el cislacio de la
Vencero el consenso de cislacio de la
Vencero en elementos
especialos



Presentación del problem Diseño de algoritmos

Algoritmo Divide y Venc Medición de tiempos studio de eficiencia

Tamaños de problen Comparación entre algoritmos Cálculo de la eficienci hibrida Errores en el cálculo constante oculta

repetidos Vectores con elemen

Vectores con eleme repetidos Presentación del problema

- 2 Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- Estudio de eficiencia
 - Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos
 - Vectores con elementos repetidos

Planteamiento

esentación del problema seño de algoritmos Algoritmo Cásicio Mgoritmo Divide y Vencera Aedición de tiempos tatollo de eficiencia Eficiencia empirica Tamaños de problema aperparea. Estado de la eficiencia bien de la eficiencia de la consecución de la Errores escultatorios con elementos petidos petidos petidos petidos carectos estados petidos p

¿Seguirá funcionando el algoritmo Divide y Vencerás si se repiten elementos en el vector?

Elementos repetidos (I)

```
ulgoritmo clásico

Ilgoritmo Divide y Venc

Redición de tiempos

tudio de eficiencia

ficiencia empirica

Tamaños de problema

Comparación entra

algoritmos

álculo de la eficiencia

Brida

Errores en el calculo de

constante constante constante
```

repetidos

Vectores con elemen
repetidos

```
    0
    1
    2
    3
    4
    5
    6

    1
    2
    2
    2
    3
    5
    5
```

Tamaño del vector: 7

Elementos repetidos (II)

```
5
          6
3
     5
          5
```

Tamaño del vector: 7

Elementos repetidos (III)

seño de algoritmos
Algoritmo Divide y Vence
Medición de tiempos
tudio de eficiencia
Eficiencia empírica
Tamaños de problema
Comparación entre
algoritmos
Tálculo de la eficiencia
Efrores en el cálculo de
Constante oculta

 0
 1
 2
 3
 4
 5
 6

 1
 2
 2
 2
 3
 5
 5

Tamaño del vector: 7

Calculamos el centro del vector. $mitad = \frac{7}{2} = 3$ (división entera)

Elementos repetidos (IV)

```
ugoritmo clásico

Ugoritmo Divide y Vencer

fedición de tiempos

tudio de eficiencia

ficiencia empírica

Tamaños de problema
Comparación entre
algoritmos

álculo de la eficiencia

librida

Errores en el cálculo de li

constante oculta
```

Tamaño del vector: 7

$$mitad = \frac{7}{2} = 3$$
 (división entera)

Elementos repetidos (V)

```
0 1 2 3 4 5 6

1 2 2 2 3 5 5

Tamaño del vector: 7
```

Al repetirse elementos, destruimos el axioma principal de la estrategia Divide y Vencerás \rightarrow El algoritmo no es válido.

Tamaño del vector: 3

Fin de la presentación