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

olseño de algoritmos
Algoritmo Clásico
Algoritmo Divide y Vencerá
Medición de tiempos
statudio de eficiencia
Eficiencia empirica
Temaños de poblema
Campasador entre
Lagoritmos
Calculo de la eficiencia
bibrida
Errores en el cálculo de la
constante oculta

- 1 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

Presentación del problema

- Presentación del problema
- - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- - Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta

El elemento en su posición

```
Presentación del problema
Diseño de algoritmos
Algoritmo Cuisico
Algoritmo Divide y Vencerá
Medición de tempos
Extudio de eficiencia empirica
Tomaños de problema
Comparación entre
algoritmos
Algoritmos
Extudio de eficiencia
habeda de la eficiencia
Emores en el cálculo de la
Constante Cuisi.
```

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

Ejemplos

Presentación del problema
Diseño de algoritmos
Algoritmo clásico
Algoritmo Divide y Vence
Medición de tiempos
Estudio de eficiencia
Eficiencia emplica
Tamalito de oproblema algoritmos
Lagoritmos
Cálculo de la eficiencia
hibrida

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

    -70
    -35
    0
    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 Vencerás
 - Medición de tiempos
- Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta

- Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta

Clásico

```
resentación del problema 
isseño de algoritmos 
Algoritmo clásico 
Algoritmo Divide y Vencerási 
Medición de tiempos 
studio de eficiencia 
Eficiencia empirica 
Dimahos de problema 
algoritmos entir 
algoritmos 
Calculo de la eficiencia 
Errores en el cálculo de la 
ectores con elementos
```

```
int elementoEnSuPosicion(const vector<int> &v) {
    for (int i = 0; i < v.size(); i++)
        if (v[i] == i)
            return i;
    return -1;
}</pre>
```

Ejemplo

```
resentación del problema

isono de algoritmos

Algoritmo Esto.

Algoritmo Esto.

Algoritmo Divide y Vencerás

Medición de tiempos

audio de eficiencia

Eficiencia empirica

Canandos de problema

comparador entre

algoritmos

biendos.

Errores en el calculo de la

constrate doute.
```

```
    0
    1
    2
    3
    4
    5
    6

    -1
    0
    2
    9
    15
    21
    66
```

Tamaño del vector: 7

```
Presentación del problema
Diseño de algoritmos
Algoritmo Cásico
Algoritmo Divide y Vencer
Medición de tiempos
Entudio de eficiencia
Eficiencia empirica
Tamaños de problema
Comparación entre
algoritmos
Calculo de la eficiencia
hibrida en el cáclulo de l
Contratité ciulia
```

```
0 1 2 3 4 5 6

-1 0 2 9 15 21 66

Tamaño del vector: 7
```

$$u v[0] = 0? \text{NO} \rightarrow \text{Continuamos}.$$

```
Presentación del problema
Piseño de algoritmos 
Algoritmo ciládo
Algoritmo Didde y Vencerá
Medición de tiempos
ictudio de eficiencia
Citado de eficiencia
Capanidos de problema
algoritmos
Cilculad de la eficiencia
Cilculad de la eficiencia
Cilculad de la eficiencia
Cincianto de problema
algoritmos
```

 $v[1] = 1? \text{NO} \rightarrow \text{Continuamos}.$

```
resentación del problema

Diesiño de algoritmos

Algoritmo Cisión

Algoritmo Divinde y Vencerás

Medición de tiempos

situadio de eficiencia

Eficiencia amplinica

Zamaños de problema
algoritmo de eficiencia

Edicinica de migrinica

Calculo de la eficiencia

Bibrida

Contacte de calculo de la

Contacte de calculo de la

Contacte de calculo de la

Contacte de calculo de la
```

Tamaño del vector: 7

$$\ \ \, \text{$\underline{\i}$} v[2] = 2? \stackrel{\textstyle \mathsf{S}}{\mathsf{I}} \to \textit{return } \mathbf{2}.$$

- Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta

Divide y Vencerás

Presentación del problema Diseño de algoritmos Algoritmo Divide y Vencerá Medición de tiempos titudo de eficiencia Eficiencia empirica Eficiencia empirica Jamaños de problema algoritmos Lagoritmos Calcidio de la eficiencia biblioria en eficiencia biblioria en eficiencia biblioria en eficiencia constante ocultar fectores con elementos epetidos

```
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);
```

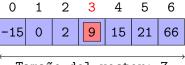
Ejemplo

```
Presentación del proteima 
Diseño de algoritmos 
Algoritmo Clásico 
Algoritmo Divide y Vencerás 
Medición de tiempos 
istudio de eficiencia 
Eficiencia empírica 
Tamaños de problema 
Comparación entre 
algoritmos 
Cálculo de la eficiencia 
hibrida
```

```
0 1 2 3 4 5 6
-15 0 2 9 15 21 66
```

Tamaño del vector: 7

resentación del problema liseño de algoritmos Algoritmo Clásico Algoritmo Divide y Vencerás Medición de tiempos studio de eficiencia Eficiencia emplicia Tamalhos de problema compazición entre Cálicito de la eficiencia hibrida



Tamaño del vector: 7

Calculamos el centro del vector. $mitad = \frac{ini + fin = 6}{2} = 3$

```
resentación del problema
isello de algoritmos
Algoritmo Divide y Vencerás
Modición de tiempos
studio de eficiencia
Eficiencia emplica
Tamallos de problema
comparación entre
algoritmos
Calcidio de la eficiencia
Expresse en el cilculos de la
Terrores en el cilculos de la
```

$$\label{eq:mitad} \begin{aligned} \textit{mitad} &= \frac{\textit{ini+fin}=6}{2} = 3\\ \textit{i.v}[3] &= 3? \ \text{NO}\\ \textit{i.v}[3] &= 9 > 3]? \ \text{Si} \rightarrow \text{exploramos desde } \textit{inicio} = 0 \ \text{a}\\ \textit{mitad} &= 1 = 2 \end{aligned}$$

```
resentación del problema

sissño de algoritmos

Algoritmo Clásico

Algoritmo Divide y Vencerás

Medición de tiempos

studio de eficiencia

Eficiencia emplica

Compazición entre

algoritmos

algoritmos

Eficiencia la eficiencia

hibrida.
```

0 1 2 3 4 5 6 -15 0 2 9 15 21 66 Tamaño del vector: 7

0 1 2 -15 0 2

Tamaño del vector: 3

```
Presentación del problema

Diseño de algoritmos

Algoritmo citásico

Algoritmo Divide y Vencerá

Medición de tiempos

Estudio de eficiencia

Eficiencia empirica

Tumaños de problema

algoritmos

Calición de la eficiencia

Eficiencia estudio

Estudio de estudio de estudio de la eficiencia

Estudio de la
```

0 1 2 3 4 5 6

-15 0 2 9 15 21 66

Tamaño del vector: 7

0 1 2

-15 0 2

Tamaño del vector: 3

$$mitad = \frac{ini + fin = 2}{2} = 1$$
 $iv[1] = 1? \text{ NO}$
 $iv[1] = 1? \text{ NO}$
 $iv[2] = 1? \text{ NO}$

```
15 21
Tamaño del vector: 7
Tamaño del vector: 3
```

- Diseño de algoritmos
 - Algoritmo clásico
 - Algoritmo Divide y Vencerás
 - Medición de tiempos
- Eficiencia empírica
 - Tamaños de problema
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta

Modificación de código fuente

resentación del proteima sieño de algoritmos lagoritmo Cásico Algoritmo Divide y Vencersi Medición de tiempos tudios de eficiencia Eficiencia empirica Tamailos de problema Comparación entre Cásiculo de la eficiencia biblida Errores en el cásiculo de la constante oculta ectores con elementos petidos problema Constante oculta ectores con elementos petidos.

```
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 Venci

Estudio de eficiencia

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

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
 - Comparación entre algoritmos
 - Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos

Presentación del problema Diseño de algoritmos Algoritmo clásico Algoritmo Divide y Vencer Medición de tiempos

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

Errores en el cálculo d constante oculta Vectores con elementos 1 Presentación del problema

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

Tamaños de problema

Algoritmo Divide y Vence
Medición de tiempos
studio de eficiencia
Eficiencia empírica
Tamaños de problema
comparación entre
algoritmos
Calculo de la eficiencia
nibrida
Errores en el cálculo de
constante oculta
errores con

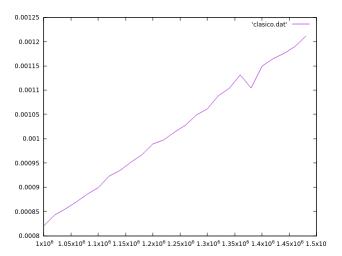
Algoritmo	Eficiencia	Tamaño inicial	Tamaño final	Incremento
Clásico	O(n)	1.000.000	1.480.000	20.000
Divide y Vencerás	O(log(n))	1.000.000	13.000.000	500.000

Algoritmo clásico

Diseño de algoritmos Algoritmo clásico Algoritmo Divide y Vene Medición de tiempos Estudio de eficiencia Eficiencia empírica

Tamaños de problema Comparación entre algoritmos Cálculo de la eficiencia tibrida Errores en el cálculo e

Errores en el cálculo de constante oculta Vectores con elementos repetidos

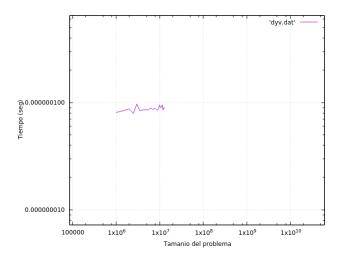


Algoritmo Divide y Vencerás

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

Cálculo de la eficienci: híbrida Errores en el cálculo constante oculta

Vectores con elementos



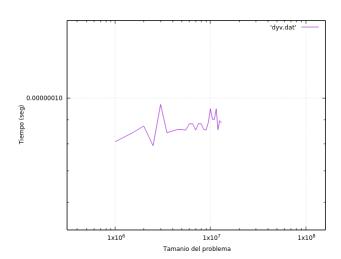
Algoritmo Divide y Vencerás (zoom)

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

Tamaños de problema Comparación entre algoritmos Cálculo de la eficiencia níbrida

constante oculta

Vectores con elementos

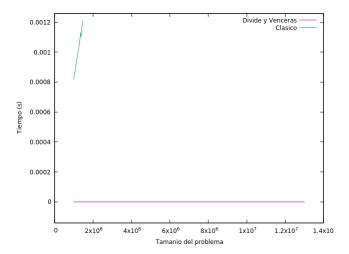


Comparación entre ambos algoritmos

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

híbrida
Errores en el cálculo de l
constante oculta

'ectores con elementos
coetidos



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

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

Vectores con elementos repetidos

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

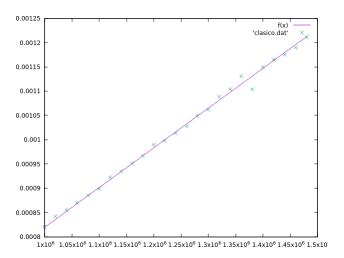
Errores en el cálculo de la constante oculta

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

Algoritmo clásico

```
sesentación del problema 
issello de algoritmos 
Algoritmo clásico 
Algoritmo Divide y Vencera 
Medición de tiempos 
studio de eficiencia 
Eficiencia empirica 
Tamalios de problema 
companya de la 
companya de la 
companya de la 
companya de la 
problema 
companya de la 
companya de la 
companya de la 
problema 
companya de la 
companya de la 
problema 
companya de la 
companya d
```

Errores en el cálculo de constante oculta Vectores con elementos



Algoritmo Divide y Vencerás

Presentación del problema

Diseño de algoritmos

Algoritmo Clasico

Algoritmo Divide y Vencerais

Medición de tiempos

Entudio de eficiencia

Eficiencia empirica

Tamalitos de problema

Comparación entre

Cicludio de la deficiencia

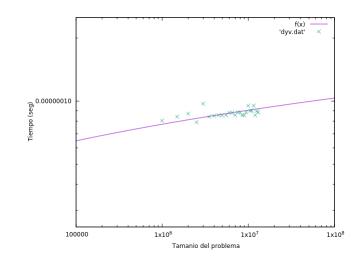
hibrida

Errores en el cálculo de la

Errores en el cálculo de la

Errores en el cálculo de la

Vectores con elementos



Presentación del problema Diseño de algoritmos Algoritmo clásico Algoritmo Divide y Vence Medición de tiempos

Tamaños de problema Comparación entre algoritmos Cálculo de la eficiencia níbrida Errores en el cálculo de

Vectores con elementos

/ectores con elementos epetidos

- 1 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
 - O Cálculo de la eficiencia híbrida
 - Errores en el cálculo de la constante oculta
- 4 Vectores con elementos repetidos

Planteamiento

resentación del problema leiseño de algoritmos Algoritmo clásico Algoritmo Divide y Vencerá Medición de tempos studio de eficiencia Eficiencia empirca Compazación entre algoritmos clásica de la eficiencia Calcula de la eficiencia Emergen en el cálculo de la constante ocuta fortose con elementos

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

Elementos repetidos (I)

```
seño de algoritmos
Algoritmo clásico
Mgoritmo Divide y Vence
Medición de tiempos
tudio de eficiencia
Eficiencia empirica
Tamaños de problema
Sigoritmos
Salculto de la eficiencia
bibrida
Errores en el cálculo de
constante contra
```

```
    0
    1
    2
    3
    4
    5
    6

    1
    2
    2
    2
    3
    4
    5
```

Tamaño del vector: 7

Elementos repetidos (II)

```
seño de algoritmos 

Ngoritmo Chisico 

Ngoritmo Divide y Vence 

Medición de tiempos 

tudio de eficiencia 

Eficiencia empírica 

Tamaños de problema 

Comparación entre 

algoritmos 

algoritmos 

Sibudio de la eficiencia 

Errores en el cálculo de 

constante oculta.
```

Vectores con element

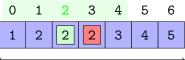
```
0 1 2 3 4 5 6

1 2 2 2 3 4 5

Tamaño del vector: 7
```

Elementos repetidos (III)

Presentación del problema
Diseño de algoritmos
Algoritmo clásico
Algoritmo Diséño de Vencer
Medición de tiempos
Estudio de eficiencia
Eficiencia empirica
Tamaños de problema
Comparación entre
Lajoritmos
Lajoritmos
Entres en el cálculo de la
constante oculta
Vectores con elementos



Tamaño del vector: 7

Calculamos el centro del vector. $mitad = \frac{ini+fin=6}{2} = 3$

Elementos repetidos (IV)

| Control | Cont

$$mitad = \frac{6}{2} = 3$$

Tamaño del vector: 7

Elementos repetidos (V)

0 1 2 3 4 5 6
1 2 2 2 3 4 5

Tamaño del vector: 7

$$mitad = \frac{6}{2} = 3 \\ 4 \quad 5 \quad 6 \\ \hline 3 \quad 4 \quad 5$$

Tamaño del vector: 3

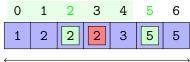
Al repetirse elementos, destruimos el axioma principal de la estrategia Divide y Vencerás \rightarrow El algoritmo no es válido.

Sin embargo...

```
esentación del problema
seño de algoritmos
Algoritmo clásico
Ugoritmo Divide y Vencerás
Medición de tiempos
audio de eficiencia
Eficiencia empírica
Tamaños de problema
```

algoritmos Cálculo de la eficiencia híbrida Errores en el cálculo de constante oculta Vectores con elementos

Vectores con elementos repetidos Hay un caso en el que puede funcionar: si en alguna iteración v[mitad] = mitad.



Tamaño del vector: 7

Tamaño del vector: 3

resentación del problema

Diseño de algoritmos

Algoritmo Cisico

Algoritmo Divide y Vencerá

Medición de tempos

Eficiencia empirca

Tamaños de problema

Compusación entre

algoritmos

algoritmos

Eficiencia de la ficiencia

bibrida

Errores en el cálculo de la

constante oculta

rectores con elementos

Fin de la presentación