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
Tamalitos de problema 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>
```

- 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

resentación del problema risorio de algoritmos Algoritmo cidado Algoritmo Cidado Algoritmo Divide y Vencersi Medición de tiempos tutido de eficianos Eficiencia empirica Tamaños de problema algoritmos algoritmos Eficiencia entre Calculo de la eficiencia Foreres en el calculo de la constante eculta constante eculta contratos con elementos prediores con elementos con elementos prediores con elementos prediores con elementos con elementos prediores con elementos con elementos prediores con elementos con elemen

```
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);
```

- 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

sesentación del proteina seño de algoritmos Ugoritmo Clásico Ugoritmo Divide y Vencerár Acclicán de tiempos tudio de eficiencia tudio de eficiencia Eficiencia empirica Tamaños de problema Comparación entre Lafelato de la eficiencia libidio de la eficiencia tudio de la eficie

```
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 Venc

Estudio de eficiencia

Lamanos de problema Comparación entre algoritmos Cálculo de la eficiencia hibrida Errores en el cálculo de l constante oculta

Vectores con elementos repetidos

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
 - 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

Vectores con elementos

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

Tamaños de problema

Algoritmo Crisico
Medición de tiempos
studio de eficiencia
Eficiencia empírica
Tanaños de problema
comparación entre
algoritmo
Calculo de la eficiencia
nibrida
Errores en el cálculo de
constante oculta
responso en contra

epetidos

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

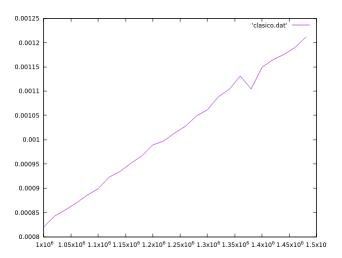
Algoritmo clásico

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

Tamaños de problem Comparación entre algoritmos Cálculo de la eficiencia úbrida Errores en el cálculo

Errores en el calculo de constante oculta Vectores con elementos repetidos

Fin de la presentació



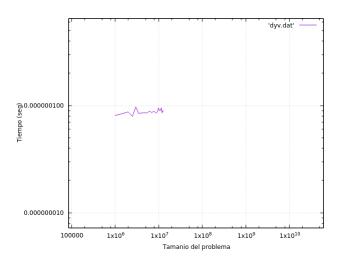
Algoritmo Divide y Vencerás

Presentación del problema
Diseño de algoritmos
Algoritmo clásico
Algoritmo Divide y Vencerás
Medición de tiempos
Eficiencia
Eficiencia empirica
Tamaños de problema

Errores en el cálculi constante oculta

Vectores con elementos

Fin de la presentación



Algoritmo Divide y Vencerás (zoom)

Presentación del problema

Diseño de algoritmos

Algoritmo clásico

Algoritmo Divide y Vencerás

Medición de tiempos

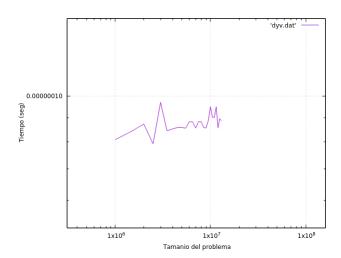
Estudio de eficiencia

Eficiencia emplifica

Tamaños de problema Comparación entre algoritmos Cálculo de la eficiencia níbrida

constante oculta

Vectores con elementos



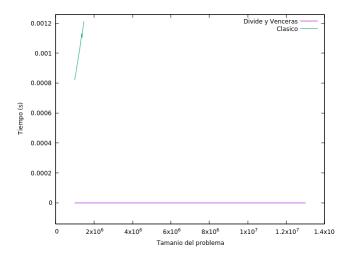
Comparación entre ambos algoritmos

Presentación del problema
Diseño de algoritmos
Algoritmo clásico
Algoritmo Divide y Vencerás
Medición de tiempos
Estudio de efficiencia
Efficiencia empírica
Tamaños de problema
Comporación entre
Ligoritmos
Eficiencia se ficiencia

híbrida
Errores en el cálculo de l
constante oculta

'ectores con elementos
coetidos

Fin de la presentación



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

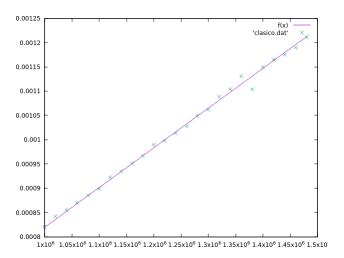
iseño de algoritmos Algoritmo clásico Algoritmo Divide y Vencera Medición de tempos studio de eficiencia Eficiencia empírica Camaños de poblema algoritmos Calculo de la eficiencia tibrida Errores en el cálculo de la companio de contra contra companio de la contra contra

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 
con 
con 
Calculo de la eficiencia 
albuida 
Efrorese en el cálculo de la 
Errorese en el cálculo de la 
Errorese en el cálculo de la 
Errorese en el cálculo de la
```

Errores en el cálculo de constante oculta Vectores con elementos

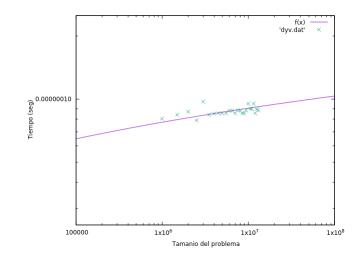


Algoritmo Divide y Vencerás

Presentación del problema
Dissiño de algoritmos
Algoritmo Cisicio
Algoritmo Disvino
Medición de tiempos
Estudio de eficiencia
Eficiencia empirica
Jamaltos de problema
algoritmo Disvino
Ciclacio de la eficiencia
Eficiencia eficiencia
Eficiencia estre
Ciclacio de la eficiencia
Eficiencia estre
Ef

ectores con elementor

Fin de la presentación



Vectores con elementos

- - 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
- Vectores con elementos repetidos

Planteamiento

resentación del problema riseño de algoritmos Algoritmo Clasico Algoritmo Divide y Vencerá Medición de tiempos studio de eficiencia Eficiencia empírica Tunaños de problema Compuención entre algoritmos Calculo de la eficiencia hibrida de ciclosido de la Contrata de contrata de Deservación de la eficiencia hibrida de ciclosido de la Contrata de cultar

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

Elementos repetidos (I)

```
iseño de algoritmos
Algoritmo Clásico
Algoritmo Chide y Vence
Medición de tiempos
studio de eficiencia
Eficiencia empírica
Tamañas de problema
Campinación entre
algoritmos
Cálculo de la eficiencia
hibrida
Errores en el cálculo de
constante coulta
```

```
    0
    1
    2
    3
    4
    5
    6

    1
    2
    2
    2
    3
    5
    5
```

Tamaño del vector: 7

Elementos repetidos (II)

```
seño de algoritmos 
Igoritmo Clásico 
Igoritmo Olvide y Vence 
ledición de tiempos 
tedición de tiempos 
tedición de tiempos 
Tamaños de problema 
Comparación entre 
algoritmos 
álculo de la eficiencia 
brida 
Errores en el cálculo de 
constante oculta
```

```
    0
    1
    2
    3
    4
    5
    6

    1
    2
    2
    2
    3
    5
    5
```

Tamaño del vector: 7

Elementos repetidos (III)

```
esentación del problema
seño de algoritmos
Ulgoritmo clásico
Ulgoritmo Divide y Vencerá
dedición de tiempos
tudio de eficiencia
ficiencia empírica
Tamalos de problema
Comparación entre
algoritmos
al
```



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

Elementos repetidos (IV)

```
essentación del problema
seño de algoritmos
Ugoritmo Olvide y Vencerá
Aedición de tiempos
tudio de eficiencia
tudio de eficiencia
fificiencia empirica
Tamaños de problema
algoritmos de entre
totales de entre
algoritmos de entr
```

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

resentación del problema riseño de algoritmos Algoritmo Cisicio. Algoritmo Divide y Vencerá Medición de tiempos studio de eficiencia Eficiencia emplica Janaños de problema Jagoritmos Jagoritmos Eficiencia de la eficiencia hibrida Errores en el cálculo de la constante oculta ectores con elementos

Fin de la presentación

Fin de la presentación