

Estrategias de programación y estructuras de datos

Grado en Ingeniería Informática

Grado en Tecnologías de la Información

Departamento de Lenguajes y Sistemas informáticos

3 Análisis de la eficiencia de los algoritmos

Coste asintótico temporal

*Javier Vélez Reyes
jvelez@lsi.uned.es*

*José Ignacio Mayorga Toledano
nmayorga@lsi.uned.es*

Departamento de Lenguajes Y Sistemas Informáticos

UNED

UNED

ETS de
Ingeniería
Informática

Análisis de la eficiencia de los algoritmos

Índice

Índice

- › Introducción
 - › ¿Qué es la eficiencia de los algoritmos?
 - › Criterios de clasificación del análisis de la eficiencia
 - › Métricas de análisis de la eficiencia de los algoritmos
- › Medida de la eficiencia de los algoritmos
 - › ¿Cómo se mide la eficiencia de los algoritmos?
 - › Medida asintótica de la eficiencia
 - › Órdenes de complejidad
- › Bibliografía

Análisis de la eficiencia de los algoritmos

Objetivos generales

Objetivos

- › Aprender qué es la eficiencia de los algoritmos
- › Aprender los criterios en torno a los cuales se puede clasificar la eficiencia
- › Conocer las principales métricas de medida de la eficiencia
- › Aprender cómo se puede calcular el coste de un algoritmo bajo 3 supuestos
 - › Medida del tiempo de ejecución
 - › Medida asintótica O
 - › Hipótesis de caso peor
- › Aprender cómo se mide la eficiencia algorítmica de
 - › Algoritmos recursivos
 - › Algoritmos iterativos
- › Adquirir una visión crítica sobre la complejidad algorítmica
- › Adquirir herramientas para seleccionar el mejor algoritmo para un problema
- › Adquirir herramientas para seleccionar la estructura más eficiente para un problema

Análisis de la eficiencia de los algoritmos

Introducción

¿Qué es la eficiencia de los algoritmos?

El análisis de la eficiencia de los algoritmos permite establecer criterios comparativos y clasificatorios entre algoritmos que ayudan a entender cómo se comportan y escalan éstos en cuanto a la consumición de recursos – tiempo y memoria fundamentalmente – con respecto a la magnitud de sus parámetros de entrada

Definición

El análisis de la eficiencia de los algoritmos es un estudio teórico, formal, comparativo e independiente de la implementación que tiene por objeto clasificar a los algoritmos en familias de complejidad de acuerdo a cómo se comportan según crece la magnitud de sus parámetros de entrada

Estudio teórico

- › Estudio formal
- › Teórico
- › Comparativo
- › Clasificadorio

Independencia

- › De recursos de máquina
- › Del lenguaje de programación
- › De aspectos constantes
- › De aspectos de escala

Análisis de la eficiencia de los algoritmos

Introducción

¿Qué es la eficiencia de los algoritmos?

El análisis de la eficiencia de los algoritmos permite establecer criterios comparativos y clasificatorios entre algoritmos que ayudan a entender cómo se comportan y escalan éstos en cuanto a la consumición de recursos – tiempo y memoria fundamentalmente – con respecto a la magnitud de sus parámetros de entrada

```
boolean ordenar (T[] v) {
    int temp;
    for (int i = 1; i < v.length; i++)
        for (int k = v.length - 1; k >= i; k--)
            if (v[k] < v[k-1]) {
                temp = v[k];
                v[k] = v[k-1];
                v[k-1]= temp;
            }
}
```

tiempo
◀

```
boolean repetidos (T[] v) {
    boolean repetidos = false;
    int i = 0;
    while (!repetidos && i < v.length)
        repetidos = contiene (v, i+1, v[i]);
    return repetidos;
}
```

```
boolean contiene (T[] v, int p, T e) {
    int i = p;
    boolean found = false
    while (!found && i <= v.length) {
        found = (v[i] == e);
        i = i + 1;
    }
    return found; }
```

Análisis de la eficiencia de los algoritmos

Introducción

¿Qué es la eficiencia de los algoritmos?

El análisis de la eficiencia de los algoritmos permite establecer criterios comparativos y clasificatorios entre algoritmos que ayudan a entender cómo se comportan y escalan éstos en cuanto a la consumición de recursos – tiempo y memoria fundamentalmente – con respecto a la magnitud de sus parámetros de entrada

Criterios de medida de la eficiencia

Criterios de medida

I. Según el tipo de recurso

La eficiencia de los algoritmos puede medirse de acuerdo a la consumición de diferentes recursos. La medida de la eficiencia de un algoritmo puede ser distinta según el recurso

II. Según el tipo de comportamiento

Fijado un tipo de recurso, la eficiencia de los algoritmos puede medirse bajo distintos supuestos en los que se supone se encuentra el problema medio a analizar

II. Según la métrica utilizada

Fijado un tipo de recurso y un determinado supuesto, debe escogerse una métrica comparativa que permita clasificar a los algoritmos en familias con un comportamiento interno similar

Espacial

Se analiza la relación funcional existente entre el tamaño del problema expresado en términos de sus parámetros de entrada y la consumición del recurso memoria

Temporal

Se analiza la relación funcional existente entre el tamaño del problema expresado en términos de sus parámetros de entrada y la consumición del recurso tiempo de cómputo

Otros

Se analiza la relación funcional existente entre el tamaño del problema expresado en términos de sus parámetros de entrada y la consumición de otros recursos como periféricos empleados, ancho de banda, consumo eléctrico, etc.

◀ Foco de atención

Análisis de la eficiencia de los algoritmos

Introducción

¿Qué es la eficiencia de los algoritmos?

El análisis de la eficiencia de los algoritmos permite establecer criterios comparativos y clasificatorios entre algoritmos que ayudan a entender cómo se comportan y escalan éstos en cuanto a la consumición de recursos – tiempo y memoria fundamentalmente – con respecto a la magnitud de sus parámetros de entrada

Criterios de medida de la eficiencia

Criterios de medida

I. Según el tipo de recurso

La eficiencia de los algoritmos puede medirse de acuerdo a la consumición de diferentes recursos. El tipo de recurso establece una partición del espacio de algoritmos distinta

II. Según el tipo de comportamiento

Fijado un tipo de recurso, la eficiencia de los algoritmos puede medirse bajo distintos supuestos en los que se supone se encuentra el problema medio a analizar

II. Según la métrica utilizada

Fijado un tipo de recurso y un determinado supuesto, debe escogerse una métrica comparativa que permita clasificar a los algoritmos en familias con un comportamiento interno similar

Peor caso

Se estima la máxima cantidad de recursos que un algoritmo puede necesitar consumir para su ejecución en función de su entrada. Supone una cota superior de estos recursos

Caso medio

Se estima el comportamiento a partir de una hipótesis sobre la distribución estadística de los datos de entrada que se supone más probable. También llamado análisis probabilístico

Mejor caso

Se estiman los recursos necesarios para su ejecución en condiciones óptimas en función de la entrada. Esta medida afecta al ejemplar más sencillo del problema

◀ Foco de atención

Análisis de la eficiencia de los algoritmos

Introducción

¿Qué es la eficiencia de los algoritmos?

El análisis de la eficiencia de los algoritmos permite establecer criterios comparativos y clasificatorios entre algoritmos que ayudan a entender cómo se comportan y escalan éstos en cuanto a la consumición de recursos – tiempo y memoria fundamentalmente – con respecto a la magnitud de sus parámetros de entrada

Criterios de medida de la eficiencia

Criterios de medida

I. Según el tipo de recurso

La eficiencia de los algoritmos puede medirse de acuerdo a la consumición de diferentes recursos. El tipo de recurso establece una partición del espacio de algoritmos distinta

II. Según el tipo de comportamiento

Fijado un tipo de recurso, la eficiencia de los algoritmos puede medirse bajo distintos supuestos en los que se supone se encuentra el problema medio a analizar

III. Según la métrica utilizada

Fijado un tipo de recurso y un determinado supuesto, debe escogerse una métrica comparativa que permita clasificar a los algoritmos en familias con un comportamiento interno similar

Cota superior. O

Se mide el coste asintótico de la ejecución de un algoritmo de acuerdo a una cota superior. Es decir, la ejecución nunca, a partir de cierto valor, sobrepasará la cota de coste establecida

Cota inferior. Ω

Se mide el coste asintótico de la ejecución de un algoritmo de acuerdo a una cota inferior. Es decir, la ejecución nunca, a partir de cierto valor, será menor que la cota establecida

Cota exacta. Θ

Se mide el coste asintótico de la ejecución de un algoritmo de acuerdo a una cota resultante de la intersección entre O y Ω . Es decir, la ejecución crece al ritmo de la cota establecida

◀ Foco de atención

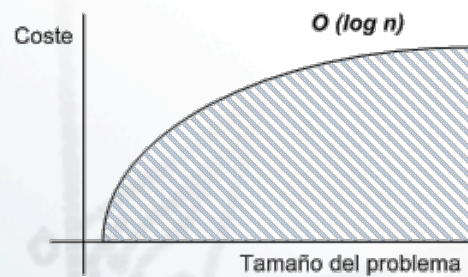
Análisis de la eficiencia de los algoritmos

Introducción

¿Qué es la eficiencia de los algoritmos?

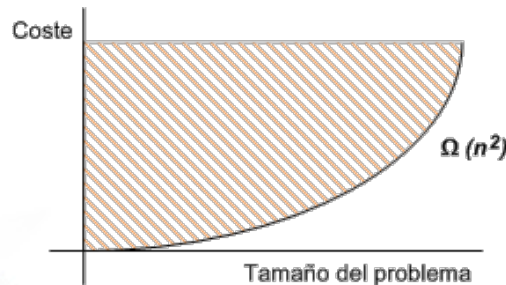
El análisis de la eficiencia de los algoritmos permite establecer criterios comparativos y clasificatorios entre algoritmos que ayudan a entender cómo se comportan y escalan éstos en cuanto a la consumición de recursos – tiempo y memoria fundamentalmente – con respecto a la magnitud de sus parámetros de entrada

Métricas de análisis de la eficiencia de los algoritmos



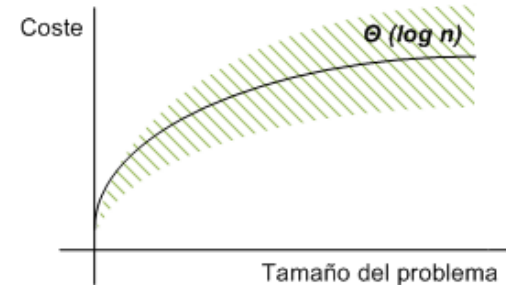
Cota superior. O

Si $g(n)$ es la función de coste de un algoritmo, se dice que $g(n)$ está en $O(f(n))$ si a partir de un cierto punto crece, a lo sumo, menos rápido que $f(n)$. Es decir, la gráfica de g está por debajo de la de f



Cota inferior. Ω

Si $g(n)$ es la función de coste de un algoritmo, se dice que $g(n)$ está en $\Omega(f(n))$ si a partir de un cierto punto crece más rápido que $f(n)$. Es decir, la gráfica de g está por encima de la de f



Cota exacta. Θ

Si $g(n)$ es la función de coste de un algoritmo, se dice que $g(n)$ está en $\Theta(f(n))$ si crece de la misma forma (al mismo ritmo) que $f(n)$

Análisis de la eficiencia de los algoritmos

Medida de la eficiencia de los algoritmos

¿Cómo se mide la eficiencia de los algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Medida asintótica de la eficiencia

Si hacemos medidas experimentales sobre sucesivas ejecuciones de un algoritmo y en cada ejecución duplicamos el tamaño del problema, observamos que el tiempo de ejecución también se duplica. En ese caso se puede decir que el coste asintótico del algoritmo es lineal

Búsqueda de un elemento en un vector

Tamaño	Tiempo medido	Tiempo de ejecución
n	$A * n + B$	t
2n	$A * 2n + B$	2t
3n	$A * 3n + B$	3t
...
kn	$A * kn + B$	kt

O (n)

El tamaño del problema es proporcional al número de elementos que contiene el vector. Sin embargo el tiempo de ejecución no se ve afectado por constantes multiplicativas o sumativas ya que el coste temporal se duplica al duplicar el tamaño del problema

Análisis de la eficiencia de los algoritmos

Medida de la eficiencia de los algoritmos

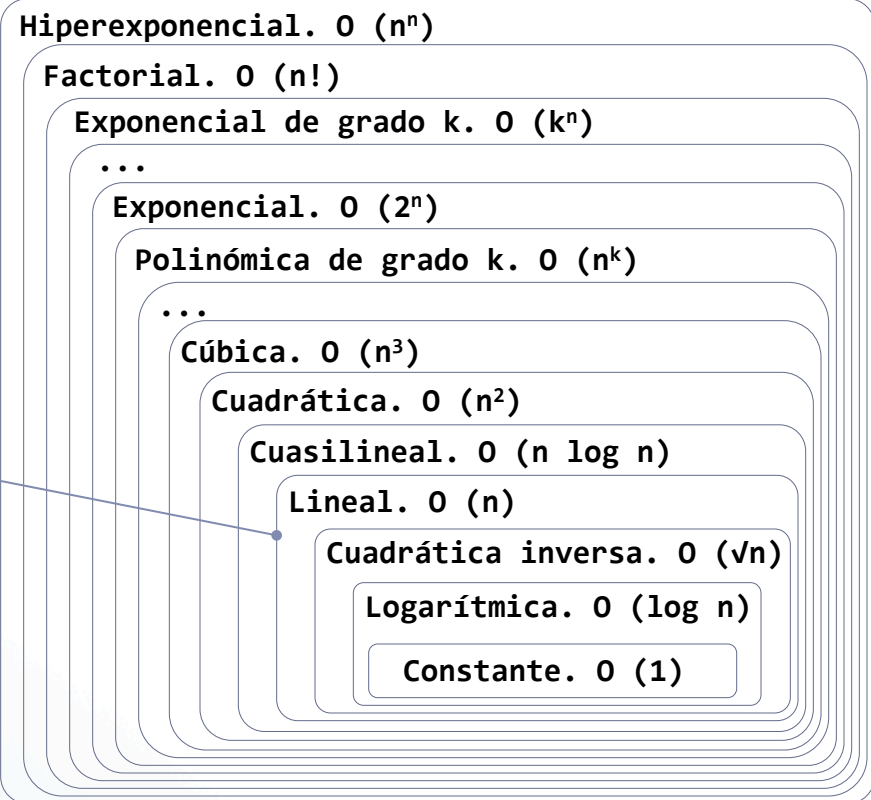
¿Cómo se mide la eficiencia de la algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Ordenes de eficiencia

El análisis asintótico nos permite clasificar el espacio de algoritmos en distintas familias que se corresponden con órdenes de complejidad diferentes

```
int factorial (int n) {  
    if (n == 0) return 1;  
    else return n * factorial (n - 1);  
}
```



Análisis de la eficiencia de los algoritmos

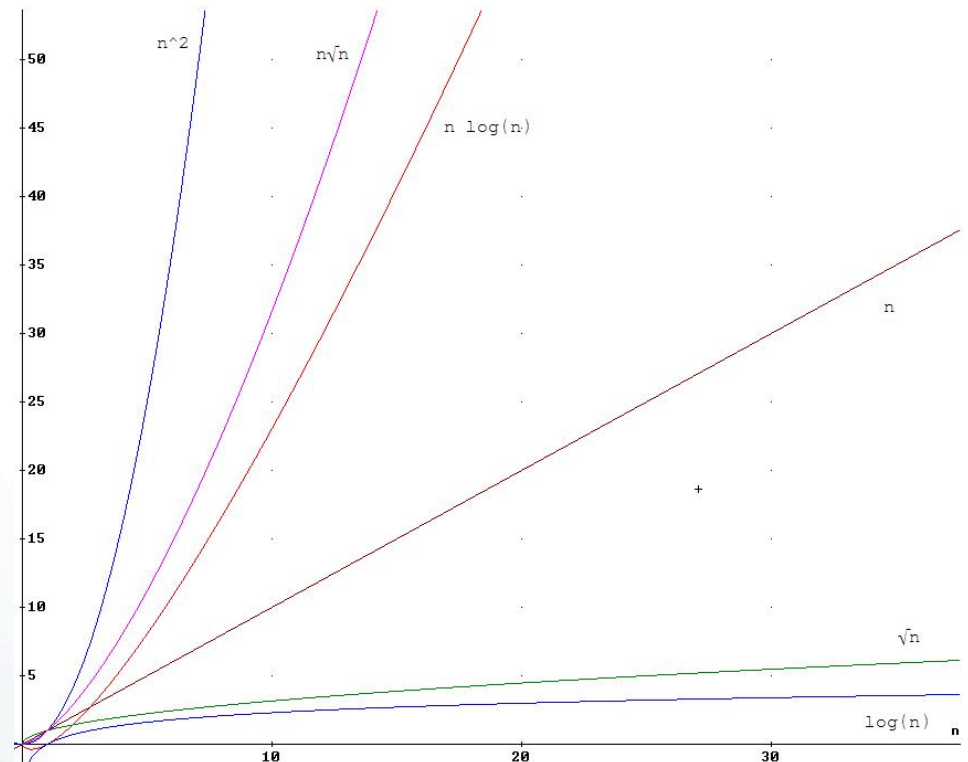
Medida de la eficiencia de los algoritmos

¿Cómo se mide la eficiencia de la algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Ordenes de eficiencia

Cada orden de eficiencia representa un nivel de crecimiento con respecto al tamaño del problema. En la grafica adjunta puede verse la relación comparativa de los niveles de crecimiento de los primeros órdenes de eficiencia anteriores



Análisis de la eficiencia de los algoritmos

Medida de la eficiencia de los algoritmos

¿Cómo se mide la eficiencia de los algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Algoritmos recursivos

El cálculo del coste temporal de un algoritmo recursivo se expresa en términos del coste de los casos base y los casos de recursión del mismo. Para ilustrar cómo se procede consideremos la función del cálculo de un número factorial en su versión recursiva. Entonces, si utilizamos $T(n)$ para representar el coste de computar la ejecución con tamaño n ...

```
int factorial (int n) {  
    if (n == 0) return 0;  
    else return n * factorial (n - 1);  
}
```

$$T(n) = \begin{cases} C_1 & \text{si } n == 0 & \text{Coste caso base} \\ T(n-1) + C_2 & \text{si } n > 0 & \text{Coste caso recurrente} \end{cases}$$

Resolución para factorial (5):

$$\begin{aligned} T(5) &= T(4) + C_2 = T(3) + C_2 + C_2 = T(2) + C_2 + C_2 + C_2 = T(1) + C_2 + C_2 + C_2 + C_2 \\ &= T(0) + C_2 + C_2 + C_2 + C_2 + C_2 = C_1 + C_2 + C_2 + C_2 + C_2 + C_2 = C_1 + 5 * C_2 \\ &= O(n) \end{aligned}$$

◀ Dado que el coste temporal resulta linealmente proporcional al tamaño del problema el orden de complejidad es lineal

Análisis de la eficiencia de los algoritmos

Medida de la eficiencia de los algoritmos

¿Cómo se mide la eficiencia de los algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Algoritmos recursivos

Para evitar en cada caso un razonamiento similar al anterior, podemos derivar una fórmula general de cálculo en función de la estructura de la función recursiva, solución analítica de la ecuación de recurrencias adjunta

$$T(n) = \begin{cases} c_1 \cdot n^k & \text{si } n < d \\ i \cdot T(n \div d) + c_2 \cdot n^k & \text{si } n \geq d \end{cases}$$

i: Número máximo de invocaciones recursivas por caso

d: Factor de decrecimiento del problema en la división del problema

n < d : Caso base

n ≥ d : Caso recursivo

$c_1 \cdot n^k$, $c_2 \cdot n^k$: Coste asociados a las partes no recursivas

k : Nivel de dependencia de las operaciones no recursivas con el tamaño del problema

0 : Constante

1 : Lineal

2 : Cuadrática

Análisis de la eficiencia de los algoritmos

Medida de la eficiencia de los algoritmos

¿Cómo se mide la eficiencia de los algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Algoritmos recursivos

Para evitar en cada caso un razonamiento similar al anterior, podemos derivar una fórmula general de cálculo en función de la estructura de la función recursiva, solución analítica de la ecuación de recurrencias adjunta

Si \oplus reducción por substracción -

$$T(n) = \begin{cases} O(n^{k+1}) & \text{si } i == 1 \\ O(i^{n/d}) & \text{si } i > 1 \end{cases}$$

$$T(n) = \begin{cases} c_1 \cdot n^k & \text{si } n < d \\ i \cdot T(n \oplus d) + c_2 \cdot n^k & \text{si } n \geq d \end{cases}$$

Si \oplus reducción por división /

$$T(n) = \begin{cases} O(n^k) & \text{si } i < d^k \\ O(n^k \log n) & \text{si } i = d^k \\ O(n^{\log_d i}) & \text{si } i > d^k \end{cases}$$

Análisis de la eficiencia de los algoritmos

Medida de la eficiencia de los algoritmos

¿Cómo se mide la eficiencia de los algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Algoritmos iterativos

El coste temporal de un algoritmo iterativo se calcula acumulativamente a partir del coste de cada una de sus instrucciones constituyentes. Estudiemos su coste

Instrucción	Coste	Instrucción	Coste
Operaciones básicas¹		Sentencias iterativas	
- Entrada / salida	$O(1)$	- for (ini;e;inc) {b}	$\max \{O(\text{ini}), n \cdot \max \{O(e), O(\text{inc}), O(b)\}\}$
- Asignación	$O(1)$	- while (e) {b}	$n \cdot \max \{O(e), O(b)\}$
- Expresiones escalares	$O(1)$	- do {b} while (e)	$n \cdot \max \{O(e), O(b)\}$
Secuencias sentencias		Invocación subprogramas	
- $s_1; s_2; s_3$	$\sum O(s_i) = \max \{O(s_i)\}$	- f (e_1, e_2)	$\text{Max} \{O(e_1), O(e_2), O(f)\}$
Sentencias condicionales		Reglas de producto y suma	
- if (e) { b_1 } else { b_2 }	$\max \{O(e), O(b_1), O(b_2)\}$	- $O(f + g)$	$O(f) + O(g) = \max(O(f), O(g))$
- switch (e) {	$\max \{O(e), O(b_1), O(b_2)\}$	- $O(f * g)$	$O(f) * O(g)$
case c_1 : b_1 ;			
case c_2 : b_2 ;			
}			

¹ Las Operaciones básicas no dependen del tamaño del problema

Análisis de la eficiencia de los algoritmos

Medida de la eficiencia de los algoritmos

¿Cómo se mide la eficiencia de los algoritmos?

La medida de la eficiencia asintótica de algoritmos pretende clasificar cada algoritmo en una familia de complejidad asintótica determinada prescindiendo de consideraciones constantes o de escala. En nuestro estudio utilizaremos medida temporal, caso peor y cota superior

Algoritmos iterativos

A partir de los análisis anteriores de coste de cada instrucción analicemos la eficiencia de un algoritmo iterativo

```
boolean ordenar (T[] v) {  
    int temp;                                O(1)  
    for (int i = 1; i < v.length; i++)      n * {  
        for (int k = v.length - 1; k >= i; k--) (n-i) * {  
            if (v[k] < v[k-1]) {              max { O(1), max {  
                temp = v[k];                  O(1),  
                v[k] = v[k-1];                O(1)  
                v[k-1]= temp;                 O(1) }  
            }  
        }  
    }  
}
```

$= O(1) + n \cdot (n-i) O(1)$
 $= \max \{ O(1) + n \cdot (n-i) O(1) \}$
 $= n \cdot (n-i) O(1) = O(n^2)$

Análisis de la eficiencia de los algoritmos

Bibliografía

Bibliografía

Bibliografía básica

Estructuras de datos en java. Weiss, Mark Allen. Pearson Addison – Wesley. ISBN 9788478290352



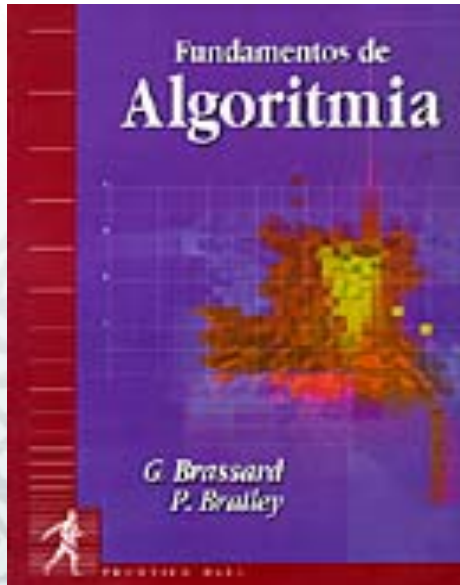
Análisis de la eficiencia de los algoritmos

Bibliografía

Bibliografía

Bibliografía complementaria

Fundamentos de Algoritmia. G. Brassard, P. Bratley. Prentice Hall. SBN: 84-89660-00-X
1997



Diseño de programas. Formalismo y abstracción. Ricardo Peña Marí. Pearson - Prentice Hall. ISBN 13: 9788420541914

