

Tema 2 Complejidad de algoritmos...



raulcb98



Fundamentos de análisis de algoritmos



1º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería
Universidad de Huelva**



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



INGENIERÍA INFORMÁTICA

FUNDAMENTOS DE ANÁLISIS DE ALGORITMOS.

TEMA 2

COMPLEJIDAD DE ALGORITMOS

RAÚL CASTILLA BRAVO
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ETSI – HUELVA)
Curso: 1º



ÍNDICE

1. Complejidad de algoritmos.	1
1.1 Coste espacial.	1
1.2 Coste temporal.	1
2. Análisis de algoritmos por conteo de operaciones elementales.	1
3. Operación básica.	2
4. Caso mejor, peor y medio.	3

1. COMPLEJIDAD DE ALGORITMOS

Para saber la cantidad de recursos que se necesitan para ejecutar un algoritmo, se estudia el tiempo que tarda en ejecutarse y el espacio que ocupa en memoria durante su ejecución. En base a esto se distinguen dos funciones de complejidad:

-Función complejidad espacial ($f_e(n)$ o $E(n)$).

-Función complejidad temporal ($f_t(n)$ o $T(n)$).

Donde n representa el tamaño del problema.

1.1 Coste espacial

Para estudiar la cantidad de memoria que ocupa un algoritmo se suman las celdas de memoria que utiliza. Para estos estudios no se distinguen entre tipos de variables: si la variable es simple ocupa una celda y si es compuesta ocupa tantas celdas como elementos que la compongan. Ahora bien, lo que si se tiene en cuenta es si las variables son estáticas o dinámicas.

1.2 Coste temporal

El tiempo que tarda en ejecutarse un algoritmo no solo depende del algoritmo en sí, sino de otros factores como: el compilador, el ordenador... Por ello, los algoritmos se escriben en pseudocódigo y se estudia el número de operaciones elementales que realiza.

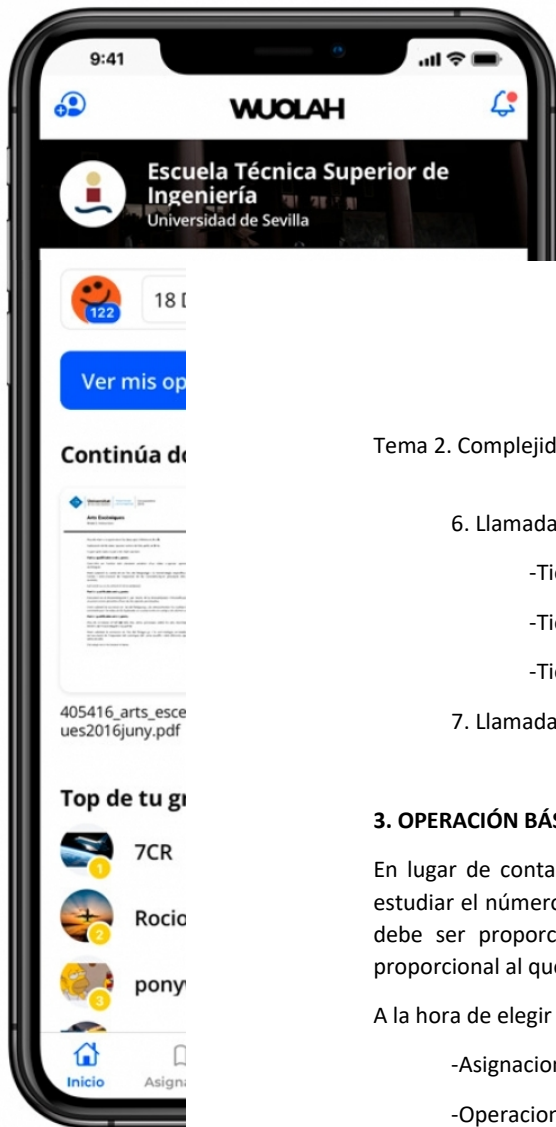
2. ANÁLISIS DE ALGORITMOS POR CONTEO DE OPERACIONES ELEMENTALES

Consideramos operaciones elementales a las siguientes:

- Operaciones aritméticas básicas.
- Asignaciones a variables de tipo predefinido.
- Saltos: llamadas o retornos a funciones o procedimientos.
- Comparaciones lógicas.
- Acceso a estructuras indexadas básicas (vectores, matrices...).

Dependiendo de la estructura con la que estemos trabajando, el conteo de operaciones se realiza de una forma u otra, pero siempre considerando el peor de los casos, es decir, el que más operaciones elementales tiene. Vamos a ver cómo se estudian las diferentes estructuras:

1. Secuencia consecutiva: suma de todas las operaciones elementales de cada instrucción.
2. Sentencia case: **caso E sea** ... $\rightarrow T = T(E) + \max\{T(S1), T(S2)\dots\}$
3. Sentencia if: **si C entonces S1 otro S2** $\rightarrow T = T(C) + \max\{T(S1), T(S2)\}$
4. Sentencia while: **mientras C hacer S** $\rightarrow T = T(C) + (n^\circ \text{ iteraciones}) * (T(S) + T(C))$
5. Sentencia for: **para ... repetir** \rightarrow Se expresa como una sentencia while.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Tema 2. Complejidad de algoritmos

6. Llamada a una función: debemos tener en cuenta tres cosas:

- Tiempo de llamada $\rightarrow T = 1$ operación elemental.
- Tiempo de evaluación de parámetros $\rightarrow T = n^{\circ}$ parámetros.
- Tiempo de ejecución de la función $\rightarrow T = T(F)$.

7. Llamada a procedimientos recursivos (tema 4).

3. OPERACIÓN BÁSICA

En lugar de contar el número total de operaciones podemos escoger una variable básica y estudiar el número de operaciones que realiza. El número de OE que realiza la variable básica debe ser proporcional al número total de OE del algoritmo, para que el resultado sea proporcional al que buscamos.

A la hora de elegir la variable básica debemos ignorar:

- Asignaciones a valores iniciales.
- Operaciones sobre variables para el control de ciclos (índices).

Algunos ejemplos de variables básicas podrían ser:

Problema	Operación básica
Búsqueda de un elemento en un conjunto.	Comparación
Ordenar un conjunto de valores.	
Recorrer un árbol.	Visitar un nodo
Resolver un sistema de ecuaciones lineales.	Suma
Multiplicar dos matrices.	Producto

Una variable básica por tanto, debe cumplir dos normas:

1. Estar relacionada con el problema a resolver.
2. Ejecutarse el mismo número de veces equivalente al número total de operaciones.

Si no hay variable básica con estas características, se elimina la primera norma y si tampoco se consigue una así; se debe hacer un análisis global.

4. CASO MEJOR, PEOR Y MEDIO

Definimos instancia como un caso específico dentro de un problema. Para estudiar el caso mejor, peor y medio de un problema de tamaño n debemos tener en cuenta:

-Las instancias del problema $\rightarrow I(n) = \{I_1; I_2; I_3; \dots; I_k\}$

-El número de operaciones de cada instancia $\rightarrow O(n) = \{O_1; O_2; O_3; \dots; O_k\}$

En base a esto, los casos que se pueden dar en la función complejidad son:

-**Caso Peor:** $T(n) = \max \{O_1; O_2; O_3; \dots; O_k\}$

-**Caso Mejor:** $T(n) = \min \{O_1; O_2; O_3; \dots; O_k\}$

-**Caso Medio:**

$$E(n) = \sum_{i=1}^k O_i P(i)$$

Donde P es la probabilidad de que ocurra cada instancia.

Para la complejidad espacial se utiliza el conjunto $C(n)$ en lugar de $O(n)$ para representar el número de celdas de memoria que ocupa el algoritmo en cada instancia.