Algoritmos y Estructuras de Datos

Tema 1
Principios del diseño de algoritmos

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

UNIVERSIDAD AUTONOMA DE MADRID
Escuela Politécnica Superior

www.cartagena99.com no hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002.
Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Parte 1

1. **Paradigmas de diseño de algoritmos**
2. **Notación Big O**
3. **Complejidad algorítmica**

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Algoritmo**
  - RAE: Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema
  - Informalmente, secuencia de instrucciones para realizar una tarea
- **Propósito**
  - Nos entrena para pensar de modo específico la resolución de ciertos problemas
  - Nos permiten descomponer problemas complejos en subproblemas más simples
- **Objetivo: *problem solving***
  - Es necesario almacenar, manejar y y recuperar datos
  - Necesitamos algoritmos (a poder ser eficientes) para resolver el

Cartagena99

CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Características deseables de un Algoritmo**
  - Ser específico
  - Con instrucciones bien definidas (sin ambigüedades)
  - Las instrucciones de un algoritmo deben poder ejecutarse en una cantidad finita de tiempo y en un número finito de pasos
  - Debe tener una entrada y una salida claras para resolver el problema
  - Cada instrucción debe ser relevante para resolver el problema

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Clasificación de los algoritmos más comunes**
  - Divide y vencerás
  - Algoritmos avaros
  - Programación dinámica

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Divide y vencerás**
  - Requiere descomponer el problema original en subproblemas más simples, resolver estos y combinar los resultados en la solución final
    - Ejemplos:
      - *Merge sort*
      - *Quick sort*
      - *Búsqueda binaria, etc.*

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Algoritmos avaros**
  - Se aplican en problemas de optimización y combinatorios. Se trata de obtener, en cada paso, la solución óptima entre varias posibles. En general se busca una solución óptima local que pueda ser, eventualmente, óptima global
  - Ejemplos:
    - *Algoritmo de Kruskal del árbol abarcador mínimo*
    - *Algoritmo de Dijkstra del camino más corto*
    - *Algoritmo de Prim*
    - *Algoritmo del problema del viajante*

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Programación dinámica**
  - Se emplea en problemas en los que los subproblemas en que se puede descomponer se *superponen* y la solución de un subproblema puede usarse en otros subproblemas
  - Ejemplo:
    - Multiplicar tres matrices: P_{20x30}, Q_{30x45}, R_{45x50}
    - $(PQ)R$ = 20x30x45 + 20x45x50 = 72000
    - $P(QR)$ = 20x30x50 + 30x45x50 = 97500

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Recursión**
  - Muy útil y usada en problemas DyV
  - Evitar bucles infinitos
    - Definir casos base
      - Determinan cuándo termina la recursión
    - Definir casos recursivos
      - Llamadas recursivas a la función hasta alcanzar algún caso base
  - Ejemplo: función factorial

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Recursión. Ejemplo

```
def factorial(n):
  # caso base. Termina la recursión
  if n==0:
    return 1
  # caso recursivo. Vuelve a llamar a la función
  else:
    f = n * factorial(n-1)
  printf(f)
  return(f)
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Búsqueda lineal (I)

- Supongamos que deseamos decidir si un número X concreto se encuentra en una cierta lista de números
- Una posible estrategia para decidir si X está en la lista es recorrer en orden los números de la lista y finalizar la búsqueda en cuanto lo encontremos o cuando se acabe la lista (y en este último caso sabremos que X no está)
- Esta estrategia se denomina **búsqueda lineal**

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Búsqueda lineal (II)

```
def searching(search_arr, x):
    for i in range(len(search_arr)):
        if search_arr [i] == x:
            return i
    return -1


>>> search_ar = [3, 4, 1, 6, 14]
>>> x=4


>>> searching(search_ar, x)
>>> print("Index position for the element x is ",searching(search_ar, x))
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Búsqueda binaria (I)

- Una alternativa más eficiente a la búsqueda lineal es la siguiente
- Supongamos que la lista de números está **ordenada**
- Entonces, una posible estrategia de búsqueda consistiría en comprobar el número que está en la mitad de la lista
  - Si el número X que buscamos es el número en la mitad de la lista, hemos acabado la búsqueda
  - Si X es anterior/menor al número en la mitad de la lista, repetimos el proceso de búsqueda en la primera mitad de la lista
  - Si X es posterior/mayor al número en la mitad de la lista, repetimos el proceso de búsqueda en la segunda mitad de la lista
- *Esta estrategia de búsqueda se denomina **busqueda_binaria***

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Búsqueda binaria (II)

```
def busqueda_binaria(numero, lista):
    l = 0
    u = len(lista)
    while (l <= u):
        pos = round((l + u) / 2)
        if (numero == lista[pos]):
            print('El número buscado está en la lista')
            return pos
        elif (numero < lista[pos]):
            u = pos - 1
        else:
            l = pos + 1
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Paradigmas de diseño de algoritmos

- **Backtracking**
  - Técnica similar a la recursión
  - Empleada en problemas de recorrido de árboles
  - Explora las ramas del árbol hasta que encuentra el objetivo o una hoja, en cuyo caso retorna (backtrack) a un nodo anterior para explorar la siguiente rama
  - Es importante, siempre que sea posible, podar aquellas ramas que no puedan dar un resultado satisfactorio en la exploración de las mismas

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Ejemplo backtracking

```
def bitStr(n,s):

 if n==1: return s

 return [digit + bits for digit in bitStr(1,s) for bits
in bitStr(n-1,s)]


print(bitStr(3,'abc'))
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Análisis del tiempo de ejecución

- **El rendimiento de un algoritmo es función**
  - Del **tamaño** de los datos de entrada
  - Del **tiempo** empleado en la ejecución
  - De la **memoria** empleada en la ejecución
- **El tiempo se mide en operaciones básicas**
  - Comparaciones, operaciones algebraicas, etc.
    - if (a == 1)
    - i = i + 1
    - a = b * c
    - etc.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Análisis del tiempo de ejecución

- **El espacio (memoria) requerido se mide en**
  - Memoria necesaria para almacenar datos en variables, constantes e instrucciones durante la ejecución
  - La memoria se puede gestionar de modo dinámico en tiempo de ejecución (reserva dinámica, pilas, etc.)
- **El tamaño de los datos de entrada**
  - Influye en el tiempo de ejecución
  - Por ejemplo, ordenar una lista de tamaño 50 frente a otra lista de tamaño 5000

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Análisis del tiempo de ejecución II

- **El tamaño de los datos de entrada**
  - Tened en cuenta las operaciones básicas (OB) que han de realizarse
    - En algoritmos de ordenación la operación básica a realizar es la **comparación**
  - A mayor número de OB mayor tiempo de ejecución
  - Obviamente, el tiempo de ejecución de un algoritmo es dependiente del Hardware en el que se ejecuta
  - Una forma independiente de evaluar el tiempo de ejecución de un algoritmo es medirlo en términos del número de OB involucradas, aunque tampoco hay una forma definitiva de cuantificar una operación pues influyen
    - El lenguaje de programación
    - El estilo de programación
    - Cómo decidimos contar operaciones

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Análisis del tiempo de ejecución III

- **Caracterizaciones de los tiempos de ejecución**
  - **Complejidad del caso peor** (*worst case complexity*). Es la cota superior de complejidad, el tiempo máximo necesario para la ejecución del algoritmo
  - **Complejidad del caso mejor** (*best case complexity*). Es la cota inferior de complejidad, el tiempo mínimo necesario para la ejecución del algoritmo
  - **Complejidad del caso promedio** (*average case complexity*). Es el tiempo promedio necesario para la ejecución del algoritmo

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Análisis del tiempo de ejecución IV

- **Ejemplo: Algoritmo *Merge Sort* de ordenación**
  - Desarrollado hace más de medio siglo
  - Muy popular en librerías de ordenación
  - Simple y eficiente
  - Es un algoritmo recursivo que emplea la estrategia DyV
    - Se trocea el problema en sucesivos subproblemas, que se resuelven de modo recursivo, combinando los resultados parciales
  - Consta de 3 pasos
    - De modo recursivo ordenar la mitad izquierda del array de entrada
    - De modo recursivo ordenar la mitad derecha del array de entrada
    - Se combinan los dos arrays ordenados en uno

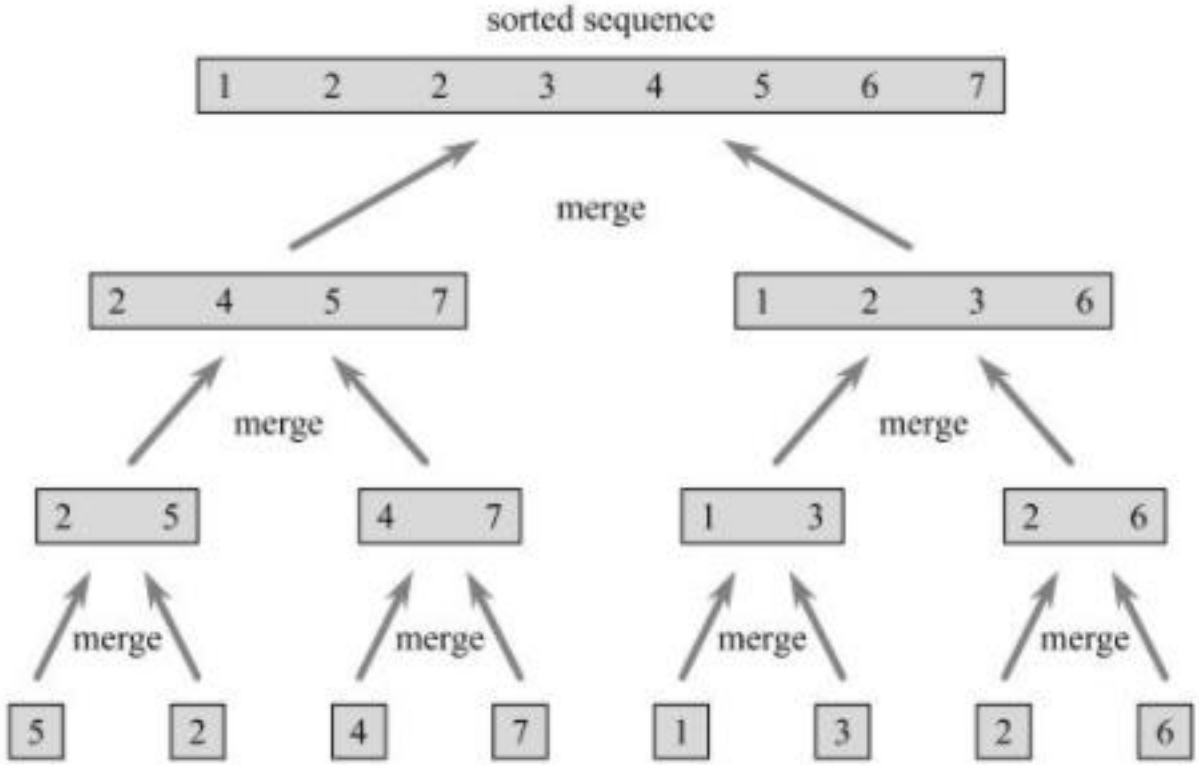
Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Análisis del tiempo de ejecución V

- **Ejemplo: Algoritmo *Merge Sort* de ordenación**



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Notación Big-O (I)

- En términos generales consideramos que el acceso a cualquier posición de la memoria RAM se realiza en la misma cantidad de tiempo, una cota superior que consideramos constante

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Notación Big-O (II)

- Queremos conocer, en términos de memoria, cómo rinde nuestro programa cuando se enfrenta a una lista grande de elementos, sabiendo que acceder a cada uno de ellos requiere de un tiempo constante, independientemente del tamaño de la lista
- Representemos el tamaño de la lista por la variable *n*
- Consideremos el tiempo promedio de acceso a un elemento de la lista de tamaño *n* como *f(n).* Entonces definimos

$$O(g(n)) = \{f \mid \exists d > 0, n_0 \in Z^+ \ni 0 \leq f(n) \leq d\, g(n), \forall n \geq n_0\}$$

  - *La clase de las funciones designadas por O de g de n consiste de las funciones f, de modo que existe un valor d mayor que 0 y un entero positivo n_0 tales que 0 es menor o igual que f de n que a su vez es menor que d veces g de n, para todo n mayor o igual a n_0*
- Si f es un elemento de la clase O(g(n)), diremos que f es O(g). g es

Cartagena99

CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Notación Big-O (III)

- El tiempo de acceso a un elemento de una lista no depende de n g(n)=1
- El tiempo promedio para acceder a un elemento de una n-lista es O(1)
- g(n)=1 es una convención, podría ser g(n)=2, etc. Es irrelevante, lo importante es considerar el crecimiento de g. Por ello se suele tomar O(1) como referencia
- Cuando una sentencia o un programa son O(1) diremos que son una sentencia o programa en **tiempo constante**, lo cual significa que no dependen de n
- La mayor parte de las operaciones de un programa son O(1)
  - Operaciones algebraicas: sumas, multiplicaciones, etc.
  - Comparación de valores
  - Asignaciones, etc.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Notación Big-O (IV)

- Se puede generalizar diciendo que f(n) = O(g(n)), O(g(n)) es el conjunto de todas las funciones con tasas de crecimiento iguales o inferiores a f(n).
  - Por ejemplo $O(n^2)$ incluye a las funciones O(n), O(nlogn), etc.

Complejidad	Nombre	Operaciones
O(1)	Constante	Añadir, acceder elemento, introducir elemento
O(logn)	Logarítmica	Buscar elemento en un array ordenado
O(n)	Lineal	Copiar, insertar, borrar, iterar
O(nlogn)	Lineal logarítmica	Ordenar (merge-sort) una lista
$O(n^2)$	Cuadrática	Camino mínimo entre dos nodos de un grafo. Bucles anidados
$O(n^3)$	Cúbica	Multiplicación de matrices

Cartagena99
CLASES PARTICULARES, TUTORIAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Notación Big-O - Clases compuestas (V)

- Cuando en un algoritmo se realizan operaciones combinadas, es necesario conocer el **tiempo de ejecución global**
- Se pueden producir diferentes escenarios
  - Operaciones secuenciales
    - Insertar en una lista O(n) y ordenar la lista O(nlogn) = O(n + nlogn) = O(nlogn)
  - Operaciones repetidas (bucles)
    - Si una operación O(f(n)) se repite O(n) veces = O(n*O(f(n))) = O(nf(n))
  - Operaciones secuenciales y anidadas combinadas
    - Si tenemos operaciones secuenciales anidadas en bucles, sumamos las complejidades de cada operación secuencial

```
n = 500                          #c0
# ejecuta n veces
for i in range(0, n):
     print(i)                    #c1
# ejecuta n veces
for i in range(0, n):
     # ejecuta n veces
     for j in range(o, n):
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Notación Big-O - Otras notaciones (VI)

- Además de la notación Big-O existen en la literatura otras notaciones para referirse a otro tipo de cotas
- **Notación Omega**: se refiere a la cota inferior del tiempo de ejecución de un algoritmo. Es decir: el caso mejor
- **Notación Theta**: se refiere al caso en que las cotas superior (Big-O) e inferior (Omega), son iguales
- Aunque ambas están definidas y desarrolladas formalmente de forma teórica, no suelen emplearse. Se suele referir normalmente la notación Big-O como notación de referencia de la complejidad algorítmica

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
- - -
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.