ESTRUCTURAS DE DATOS Y ANÁLISIS DE ALGORITMOS

- Propósitos del Curso
- Horario y Docente
- Contenido de la Asignatura
- Bibliografía
- Evaluaciones

Propósitos del Curso

•Desarrollar el pensamiento algorítmico basado en alcanzar la máxima eficiencia.

•Desarrollar la capacidad de evaluar algoritmos y proponer sus mejoras.

•Identificar las estructuras de datos que permitan la mejor implementación a la solución de un problema.

Menú Principal

Horario

- Jueves 6:00-8:00 p.m. Sábado 6:00 a.m.-8:00 a.m.
- Horario de consulta: Martes 8:00 p.m.-9:00 p.m.

- **DOCENTE : RONEY A. SUAREZ**

suarez@correo.uis.edu.co

INGENIERO DE SISTEMAS UIS

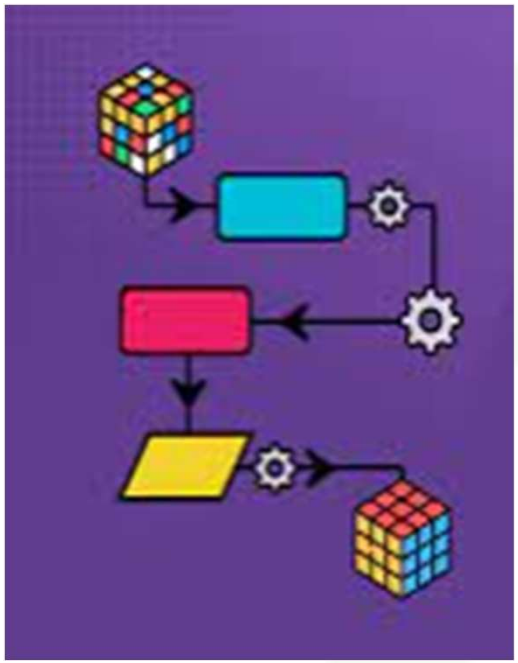
Esp. Gerencia Servicios de Salud UDES

Magister Software Libre UNAB-UOC

Menú Principal

Contenido

Menú Principal



CAPITULO 1

Análisis Algorítmico Básico

Contenido

Menú Principal Contenido

1.1. Introducción

Algoritmo

Conjunto de reglas ordenadas de manera lógica que buscan la solución de un problema.

> Conjunto de líneas de código que a partir de unas entradas de datos, realizan operaciones de consulta y/o procesamiento, para generar una salida, consumiendo unos recursos de Hardware.

https://www.youtube.com/watch?v=b3fwA3EWCd8

Criterios de Evaluación

1. Menor consumo de recursos.
2. Velocidad de generación de la salida.
3. Facilidad de Programación.
4. Longitud en líneas de código.
5. Tolerancia a fallas y/o volumen de entradas.
6. Escalabilidad

Criterio Empresarial
Eficiencia a nivel de rapidez en la generación de salidas, solución del requerimiento, costo y menor consumo de recursos.

https://www.youtube.com/watch?v=b5dhq3dSG2k

Menú Capítulo

1.2. Noción de Tiempo y Espacio

Complejidad Temporal
Evaluación en función del tiempo de ejecución del algoritmo.

Complejidad Espacial
Evaluación en función de la memoria utilizada por el algoritmo durante su ejecución.

⇨ ***Menú Capítulo***

1.3. Análisis de mejor caso, promedio y peor caso

Mejor Caso

El algoritmo se desarrolla bajo condiciones ideales.

Promedio

El algoritmo se desarrolla bajo condiciones típicas.

Peor Caso

El algoritmo se desarrolla bajo condiciones extremas.

Menú Capítulo

1.3. Análisis de Eficiencia Temporal

Se calcula de acuerdo al número de repeticiones de la operación básica como función del tamaño de la entrada.

$$T(n)=copC(n)$$

T:tiempo de ejecución
n: tamaño de la entrada
cop: tiempo de ejecución de la operación básica
C(n):cantidad de veces que se ejecuta la operación básica

Menú Capítulo

1.4. Análisis Asintótico

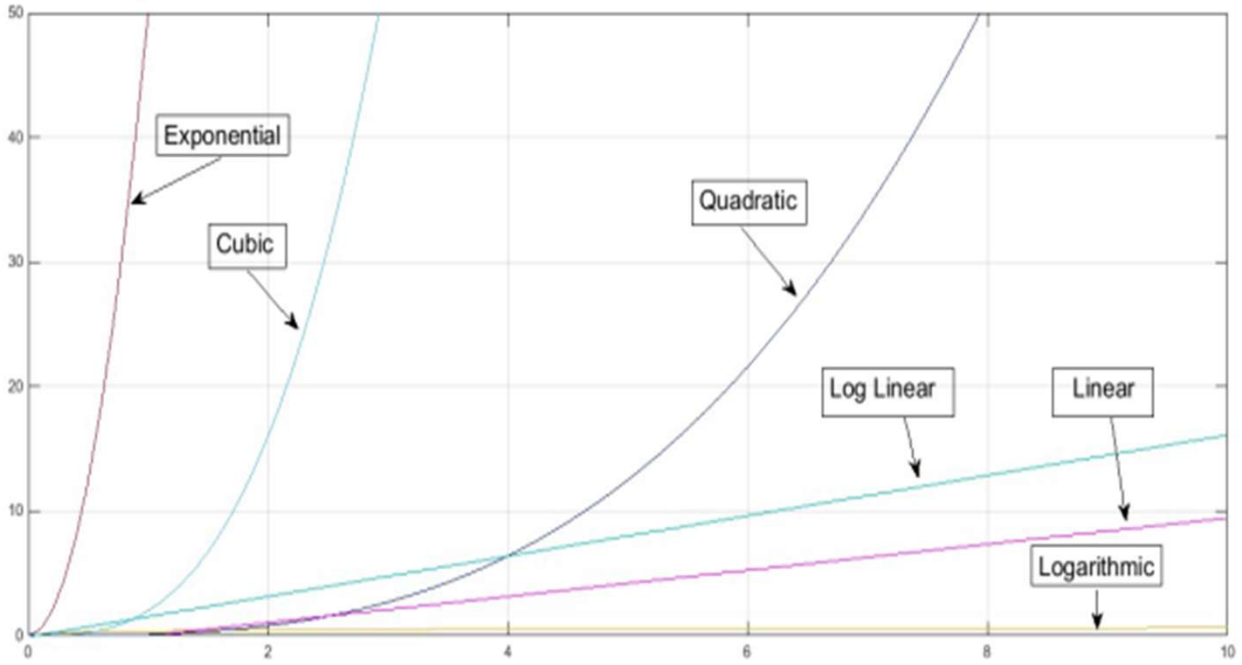
Evaluación del algoritmo en función del tamaño de su entrada. Medición del tiempo y/o espacio de acuerdo a la entrada.

f(n)=Función que representa la tasa de crecimiento de la velocidad de respuesta de un algoritmo frente a una Entrada.

Principio de Invarianza
Dos implementaciones distintas de un mismo algoritmo f1(n) y f2(n) no diferirán en su eficiencia en más de una constante multiplicativa.

1.4. Análisis Asintótico

El algoritmo se compara con otros patrones ya conocidos de modelos matemáticos.



Menú Capítulo

1.5. Notaciones

O Grande

El algoritmo se compara con otros patrones ya conocidos de modelos matemáticos buscando definir una cota superior.

Análisis de Cota Superior

La gráfica de nuestro algoritmo siempre va a estar por debajo de una función patrón g(n).

f(n) es del orden O(g(n))

Menú Capítulo

1.4. Notaciones

O Grande

Ordenes de Crecimiento Comunes

Orden	Nombre
$O(1)$	Constante
$O(log(n))$	Logarítmico
$O(n)$	Lineal
$O(nlog(n))$	nlog(n)
$O(n^2)$	Cuadrático
$O(n^3)$	Cúbico
$O(n^k),\ k > 3$	Polinomial
$O(2^n)$	Exponencial

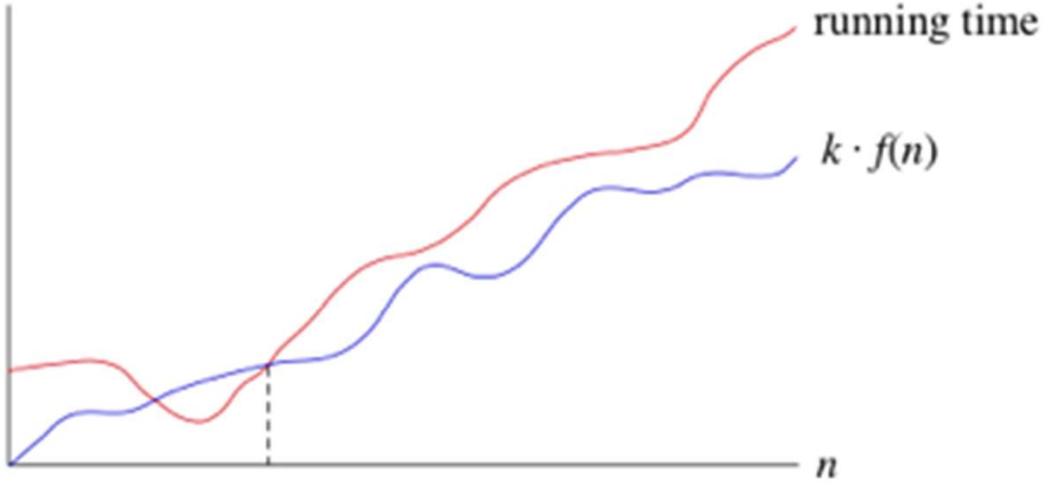
Menú Capítulo

1.5. Notaciones Omega

Análisis Límite Inferior (Cota inferior asintótica)

La gráfica de nuestro algoritmo siempre va a estar por encima de una función patrón f(n).

T(n) es del orden Ω(f(n))



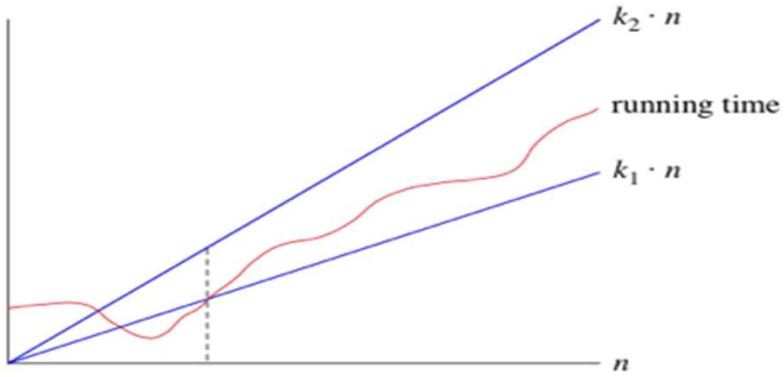
Menú Capítulo

1.5. Notaciones Theta

Análisis Ajustado(Cota inferior asintótica)

La gráfica de nuestro algoritmo siempre va a estar por encima y debajo de una función patrón f(n) ajustada por dos constantes k1 y k2.

f(n) es del orden θ(g(n))



Menú Capítulo

1.4. Análisis Asintótico

Evaluación del algoritmo en función de los ciclos de CPU desarrollados en su ejecución.

```
for i in range(n)
    p=int(input("digite un valor"))
    a=a+p
```

Tiempo de ejecución es f(n)=n

```
for i in range(n):
    p=int(input("digite un valor"))
    for j in range(p):
        a=a+p
```

Tiempo de ejecución es n x n = $f(n)=n^2$

1.4. Análisis Asintótico

```
b=0
a=0
for i in range(n):
    p=int(input("digite un valor"))
    b=b+p

for i in range(n):
    p=int(input("digite un valor"))
    for j in range(p):
        a=a+p
x=a-b
```

Tiempo de ejecución es $f(n)=n + n^2$

CAPITULO 2

Estrategias Algorítmicas Fundamentales

Contenido

Menú Principal Contenido

2.1. Algoritmos de Fuerza Bruta

Encuentran la solución a un problema basados en la evaluación de todas las posibles soluciones al mismo.

Encontrar los divisores de un número.
Búsqueda de una cadena de texto en una frase.
Encontrar una clave.

https://www.youtube.com/watch?v=dxjL_0HJg8U

Menú Capítulo

2.2. Recursividad

Cuando el algoritmo se llama a sí mismo en su implementación.

Es obligatorio que exista un caso base para evitar un ciclo infinito.

Clases

- Directa o Indirecta
- Simple o Lineal y Múltiple
- Final o No Final

Menú Capítulo

2.3. Algoritmos Ávidos

Seleccionan paso a paso la mejor decisión a partir de un grupo de candidatos, buscando obtener la solución al problema.

Encontrar la mejor ruta.

Menú Capítulo

2.4. Algoritmos Divide y Vencerás

Subdividen el problema en problemas más pequeños y así sucesivamente hasta llegar a un elemento base; con la combinación de estos básicos se organiza la respuesta.

Ordenamiento: Quicksort, Mergesort
Matemáticas: Euclides MCD

Menú Capítulo

2.5. Retroceso o vuelta atrás

Recorren todas las opciones posibles de solución tratando de cumplir una restricción, si la solución planteada incumple la restricción, se regresa de nuevo a iniciar la búsqueda.

Ocho reinas.
Búsqueda con retroceso.

Menú Capítulo