



Algoritmia

Facultad de Informática

August 8, 2024





Indice

1. Algoritmos y Algoritmia
Características
2. Comparación de Algoritmos
Recursos
3. Índices de crecimiento
Tiempos





Tabla de contenidos

1. Algoritmos y Algoritmia

Características

2. Comparación de Algoritmos

Recursos

3. Indices de crecimiento

Tiempos





Algoritmo

Algunos conceptos

Knuth - Conjunto ordenado y finito de pasos, con 1 entrada y salidas

- Independiente del lenguaje de programación
- Eficaz (se debe realizar en un tiempo finito)
- Determinista (salvo los probabilistas y concurrentes)
 - Su solución se presenta como un **modelo matemático**
 - Para demostrar qué es correcto y eficiente
- ¿Qué analizar de diferentes algoritmos?
 - Facilidad para entender y modificar
 - Usar la menor cantidad de recursos - Tiempo/espacio





Algoritmo y algoritmia

Algoritmo

Es un **conjunto prescrito de instrucciones** o reglas bien definidas, **ordenadas y finitas** que permite realizar una actividad mediante pasos sucesivos que no generan duda a quien deba realizar dicha actividad.

Dados un estado inicial y una entrada, siguiendo los pasos sucesivos, se llega a un estado final y se obtiene una solución.

Algoritmia

Es la ciencia que estudia los **algoritmos**, sus propiedades y eficiencia.



Algoritmia

Ciencia de los algoritmos

Objetivo: Desarrollo de métodos y técnicas para el diseño de algoritmos y estructuras de datos eficientes y su análisis.

- Cómo lo trata? A través de **Análisis de Eficiencia**
- Propiedades
 - Correctitud: Finalización del algoritmo y obtención del resultado deseado.
 - Eficiencia: Cantidad de recursos que se utilizan.

Diseño de Algoritmos (4Año LCC) se tratarán técnicas algorítmicas básicas como el método voraz (greedy method), la programación dinámica, los flujos sobre redes, la aleatorización.



Tabla de contenidos

1. Algoritmos y Algoritmia
Características
2. Comparación de Algoritmos
Recursos
3. Indices de crecimiento
Tiempos



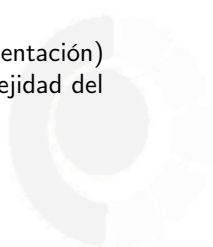


Comparación de Algoritmos

Recursos

Si hay mas de un algoritmo para resolver un problema dado, con cual nos quedamos?

- Uso de Recursos **computacionales**:
 - Tiempo de ejecución
 - Espacio de memoria
- Uso de Recursos **no computacionales**:
 - Esfuerzo de desarrollo (análisis, diseño e implementación)
calidad del código, bibliotecas utilizables, complejidad del algoritmo.





Solución de Algoritmos

Recursos

Se puede presentar como un modelo matemático, a partir de herramientas matemáticas se puede demostrar

- Correctitud
- Eficiencia

Qué se debe analizar?

- El conjuntos de casos o dominio de datos de entrada .
- Tipos de soluciones
 - Exactas
 - Uso de Recursos computacionales y no
 - Aproximadas
(cuando no se puede resolver con exactitud para la mayoría de los casos)





Solución de Algoritmos

Recursos

La eficiencia se suele medir con dos parámetros

- **Espacio**, cantidad de memoria utilizada
- **Tiempo**, lo que tarda en ejecutarse
 - Real o Empírico
 - Teórico: Función que acote (por arriba o por abajo) el tiempo de ejecución del algoritmo para ciertos valores entrada.





Tiempos

Tiempo Empírico

- Cantidades de datos: probar 100, 1000, 10000, 50000, 100000
- 1.000.000 nanosegundos = 1 milisegundo)
- 1 Nanosegundos = 1.0×10^{-6} Milisegundos
- No considerar tiempos de generación o carga de datos

```
long t1 = System.nanoTime();
```

```
...
```

```
long t2 = System.nanoTime() - t1;
```

```
System.out.println(" Tiempo real : " + t2/1e6 + " ms");
```



Tabla de contenidos

1. Algoritmos y Algoritmia
Características
2. Comparación de Algoritmos
Recursos
3. Indices de crecimiento
Tiempos





Tiempos de Ejecución

Tipos de Análisis

❶ Mejor Caso, $T_{mejor}(n)$ no fiel

Eficiencia algorítmica

Describe propiedades relacionadas con la cantidad de recursos utilizados por el algoritmo. En el caso medio (promedio) se puede considerar la situación típica de ejecución del algoritmo.



Tiempos de Ejecución

Tipos de Análisis

- 1 Mejor Caso, $T_{mejor}(n)$ no fiel
- 2 Caso Medio, $T_{prom}(n)$ más fiel

Eficiencia algorítmica

Describe propiedades relacionadas con la cantidad de recursos utilizados por el algoritmo. En el caso medio (promedio) se puede considerar la situación típica de ejecución del algoritmo.



Tiempos de Ejecución

Tipos de Análisis

- 1 Mejor Caso, $T_{mejor}(n)$ no fiel
- 2 Caso Medio, $T_{prom}(n)$ más fiel
- 3 Peor Caso, $T_{peor}(n)$ nos aseguramos

Eficiencia algorítmica

Describe propiedades relacionadas con la cantidad de recursos utilizados por el algoritmo. En el caso medio (promedio) se puede considerar la situación típica de ejecución del algoritmo.



Tiempos de Ejecución

Tipos de Análisis

- 1 Mejor Caso, $T_{mejor}(n)$ no fiel
- 2 Caso Medio, $T_{prom}(n)$ más fiel
- 3 Peor Caso, $T_{peor}(n)$ nos aseguramos
- 4 **Análisis Probabilístico, $T_{prob}(n)$ más procesamiento**

Eficiencia algorítmica

Describe propiedades relacionadas con la cantidad de recursos utilizados por el algoritmo. En el caso medio (promedio) se puede considerar la situación típica de ejecución del algoritmo.



Tiempos de Ejecución

Tipos de Análisis

- 1 Mejor Caso, $T_{mejor}(n)$ no fiel
- 2 Caso Medio, $T_{prom}(n)$ más fiel
- 3 Peor Caso, $T_{peor}(n)$ nos aseguramos
- 4 Análisis Probabilístico, $T_{prob}(n)$ más procesamiento
- 5 Análisis Amortizado, $T_{amor}(n)$ más real

Eficiencia algorítmica

Describe propiedades relacionadas con la cantidad de recursos utilizados por el algoritmo. En el caso medio (promedio) se puede considerar la situación típica de ejecución del algoritmo.



Complejidad de operaciones

Amortizado

- Hacerse una taza de cafe en el curso
 - Hay cafe
 - No hay cafe





Complejidad de operaciones

Análisis amortizado

Análisis amortizado de complejidad

Forma de medir el costo de ciertas operaciones.

- No contamos el tiempo real de cada operación
- Medimos el tiempo total T que toma realizar n operaciones
- Cada operación tiene una complejidad amortizada de $O(T/n)$