

Análisis de Algoritmos

Introducción

Contexto

- Ya podemos hacer la asociación
 - Problema computacional \Leftrightarrow MTD
 - Entrada \Leftrightarrow Instancia del problema comput.
 - Problemas computables \Leftrightarrow R

Contexto

- Ya podemos hacer la asociación
 - Problema computacional \Leftrightarrow MTD
 - Entrada \Leftrightarrow Instancia del problema comput.
 - Problemas computables \Leftrightarrow R
 - Complejidad \Leftrightarrow “Análisis” de R
 - Computable \neq “Tratable”
 - MTD \Leftrightarrow Algoritmo

Contexto

- Ya podemos hacer la asociación
 - Problema computacional \Leftrightarrow MTD
 - Entrada \Leftrightarrow Instancia del problema comput.
 - Problemas computables \Leftrightarrow R
 - Complejidad \Leftrightarrow “Análisis” de R
 - Computable \neq “Tratable”
 - MTD \Leftrightarrow Algoritmo
 - Problema computable \Leftrightarrow Infinitas MTD (alg.)
 - MTD \Leftrightarrow Cantidad de pasos (δ de transición)

Contexto

- De la clase inicial

Temas:

Algoritmos (específico) \implies Algoritmia. Análisis,
Notación Asintótica

Computabilidad

Complejidad

}

\implies Problemas (general)

Contexto

- De la clase inicial

Temas:

Algoritmos (específico) \implies Algoritmia. Análisis, Notación Asintótica

Computabilidad

Complejidad

} \implies Problemas (general)

Algebraico/Matemático
(cardinalidad)

Algorítmico/MT

Contexto

- De la clase inicial

Temas:

Algoritmos (específico) \implies Algoritmia. Análisis, Notación Asintótica

Computabilidad

Complejidad

} \implies Problemas (general)

Clases de
complejidad/MT

Algebraico/Matemático
(cardinalidad)
Algorítmico/MT

Algoritmia

Temas:

Algoritmos (específico) ==> Algoritmia. Análisis,
Notación Asintótica

Algoritmia

Temas:

Algoritmos (específico) \implies Algoritmia. Análisis,
Notación Asintótica

Algoritmia

Diseño (creación-propuesta)

Análisis (comportamiento)

} \implies De algoritmos

Algoritmia

- Diseño: lo han visto hasta este punto en múltiples asignaturas
- Análisis de Algoritmos (Leiserson): "The theoretical study of computer-program performance and resource usage"
- Análisis de Eficiencia: Cantidad de recursos que se utilizan (tiempo, espacio)

Algoritmia

- ¿Por qué hacer análisis de algoritmos?
 - Comparación teórica
 - Escalabilidad (cuánto más grande se puede, hasta dónde llego)
 - Matemática algorítmica provee un lenguaje de comportamiento de programas
 - Análisis de t y e se pueden relacionar en algunos casos

Algoritmia

- El análisis teórico está apoyado por el principio de invarianza:
 - "Según el principio de invarianza, dos implementaciones diferentes del mismo algoritmo no difieren en tiempo de ejecución más que en una constante multiplicativa"
- Relación directa con
 - Sea M una MTD con k cintas, decimos que M es de complejidad $t(n)$, si para toda entrada de longitud n , M hace a lo sumo $t(n)$ mov.

Tiempo y Espacio

- Análisis de t y e
 - Dado un algoritmo A , el tiempo de ejecución $t_A(n)$ de A es la cantidad de pasos, operaciones o acciones elementales que debe realizar el algoritmo al ser ejecutado en una instancia de tamaño n .
 - El espacio $e_A(n)$ de A es la cantidad de datos elementales que el algoritmo necesita al ser ejecutado en una instancia de tamaño n , sin contar la representación de la entrada

Análisis de Algoritmos

- Definiciones de $t_A(n)$ y $e_A(n)$ son ambiguas en dos sentidos:
 - No especifica claramente cuáles son las operaciones o datos elementales.
 - Dado que existe más de una instancia de tamaño n no está claro cuál de ellas es la que se tiene en cuenta para el análisis (o cuáles, si son relevantes).

Análisis de Algoritmos

- Operaciones Elementales:
 - Tiempo de ejecución está acotado por una constante que depende sólo de la implementación usada
 - No depende del tamaño de la entrada
 - Sólo interesa el número de operaciones elementales
 - En general, las sumas, multiplicaciones y asignaciones son operaciones elementales

Análisis de Algoritmos

- Tipos de instancias (¿Por qué) - Tipos de análisis (si hay dependencia de datos):
peor-mejor-promedio-probabilístico
- Relación con
 - Sea M una MTD con k cintas, decimos que M es de complejidad $t(n)$, si para toda entrada de longitud n , M hace a lo sumo $t(n)$ mov.

Análisis de Algoritmos

- Tipos de instancias (¿Por qué) - Tipos de análisis (si hay dependencia de datos): peor-mejor-promedio-probabilístico
- Relación con
 - Sea M una MTD con k cintas, decimos que M es de complejidad $t(n)$, si para toda entrada de longitud n , M hace **a lo sumo $t(n)$** mov.

Análisis de Algoritmos

- Notación Asintótica
 - Utilizaremos la noción de tiempo ignorando las constantes multiplicativas que lo afectan
 - Caracteriza en forma simple la eficiencia (o uso de recursos) de los algoritmos
 - Se independiza de las características específicas de la implementación
 - Análisis del comportamiento de las funciones en el límite, considerando sólo su tasa de crecimiento (notación asintótica)

Análisis de Algoritmos

- (1) Estructuras de control
 - (2) Barómetro
 - (3) Análisis del caso promedio
 - (4) Análisis amortizado
 - (5) Recurrencias
 - (6) No veremos análisis asociados a algoritmos específicos *de* estructuras de datos
 - (7) Diseño + análisis
- Usualmente aplicado a peor caso, son “en detalle”
- Usualmente dependiente del “tipo de entrada”
- ¿? Un tipo específico de algoritmos
- Greedy
 - Divide-and-Conquer
 - Dynamic programming
 - Algoritmos probabilísticos