



Problemas tratables e intratables

Héctor Alejandro Montes

h.a.montes@fi.uaemex.mx

<http://fi.uaemex.mx/h.a.montes>

Advertencia

No use estas diapositivas como referencia única de estudio durante este curso. La información contenida aquí es sólo una guía para las sesiones de clase y de estudio futuro. Para obtener información más completa, refiérase a la bibliografía dada durante la presentación del curso.

Complejidad de un Problema

- ¿Cómo saber que un problema es *difícil* o *complejo*?
- ¿Cuándo decimos que es un problema es *difícil* o *complejo*?
- Complejidad:
 - Algorítmica
 - Clasificación
- Los problemas se conocen *Tratables* o se sospechan *Intratables*

Problemas Tratables e Intratables

- **Problema Tratable:** Se puede resolver por un algoritmo en tiempo polinomial
 - La cota superior (*upper bound*) es **polinomial**
- **Problema Intratable:** No se puede resolver por un algoritmo en tiempo polinomial
 - La cota inferior (*lower bound*) es **exponencial**

Complejidad algorítmica o computacional

- Funciones *polinomiales* y *exponenciales*

constant	$O(1)$
logarithmic	$O(\log n)$
linear	$O(n)$
n-log-n	$O(n \times \log n)$
quadratic	$O(n^2)$
cubic	$O(n^3)$
exponential	$O(k^n)$, e.g. $O(2^n)$
factorial	$O(n!)$
super-exponential	e.g. $O(n^n)$

Determinismo y No-determinismo

- Algoritmo *determinista*
 - En cualquier paso del algoritmo, sólo hay un siguiente paso posible
- Algoritmo *no-determinista*
 - Puede existir más de un paso a elegir y no siempre se elegirá el mismo

Clasificación: P

- ¿Se pueden resolver todos los problemas en tiempo polinomial?
 - No. Existen muchos problemas que no se pueden resolver, no importa el tiempo involucrado
- Problemas para los que se ha encontrado un algoritmo determinista que corra en tiempo polinomial: Clase **P**
 - Una larga lista de problemas de uso práctico están en P

Clasificación: NP

- Problemas para los que **no** se ha encontrado un algoritmo determinista que corra en tiempo polinomial: Clase **NP**
- Una pregunta abierta en investigación:
 - **P = NP?**

Clase NP-Completo

- Conjunto de problemas que se han probado en NP
- Si existe un algoritmo que transforme cualquier problema en NP a otro también en NP, ese otro problema es *NP-Completo*
 - La transformación debe ser de orden polinomial
- Son problemas de decisión
- Cualquier solución al problema puede ser verificada rápidamente
- Muchos problemas de interés son *NP-Completo*s

Problemas

- *El problema del agente viajero*
- El problemas de programación de procesos (*job-shop scheduling*)
- El problema de satisfactibilidad (3-SAT)
- The *Bin Packing* problem
- El problema de horarios (timetable)
- Problema “El dilema del prisionero”

NP-Hard

- Problemas en NP
- Son “al menos tan difíciles como los NP-Completos”
- A veces asociados con los problemas de optimización

Clasificación general

