Introducción a la Algoritmia

IIC2283 – Diseño y Análisis de Algoritmos

Diego Arroyuelo, Juan P. Castillo
diego.arroyuelo@uc.cl

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

2025-2

¿A qué nos dedicamos en ciencia de la computación?

Construcción de algoritmos

- Diseño de algoritmos eficientes para resolver distintos tipos de problemas
- Demostración de cotas inferiores para el tiempo (u otro recurso relevante como el espacio ocupado) necesario para solucionar un problema
  - Demostración de que hay problemas que no pueden ser resueltos de manera eficiente
- Implementación eficiente de los algoritmos diseñados en un modelo de computación

¿Qué es un algoritmo?

Definición Informal

Un **algoritmo** es una secuencia finita, ordenada, y no ambigua de instrucciones (o pasos a seguir) que permiten resolver un problema.

Resolver un problema significa que para cada una de sus instancias obtenemos la solución correspondiente

Un algoritmo recibe una instancia del problema como entrada y produce su correspondiente solución como salida

¿Qué es un algoritmo?

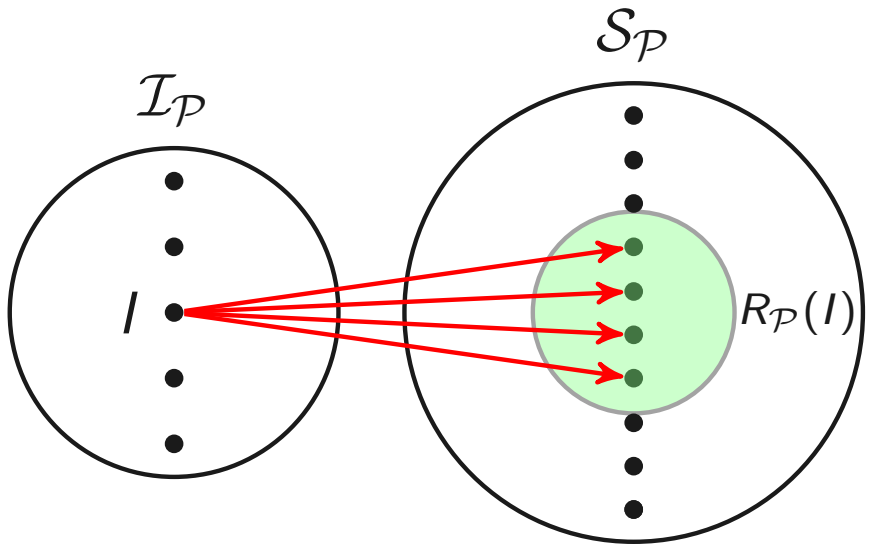
Definición Formal

Para un problema abstracto $\mathcal{P}$, sea:

- $\mathcal{I}_\mathcal{P}$ el conjunto, posiblemente infinito, de todas las instancias de $\mathcal{P}$.
- $\mathcal{S}_\mathcal{P}$ el conjunto de soluciones a todas las instancias de $\mathcal{P}$.
- $R_\mathcal{P} \subseteq \mathcal{I}_\mathcal{P} \times \mathcal{S}_\mathcal{P}$ la relación binaria entre instancias de $\mathcal{P}$ y sus soluciones.
- $R_\mathcal{P}(I) = \{s \mid (I, s) \in R_\mathcal{P}\}$ es el conjunto de soluciones de una instancia $I \in \mathcal{I}_\mathcal{P}$.

Formalmente, todo algoritmo $\mathcal{A}_\mathcal{P}$ que resuelve el problema $\mathcal{P}$ es una representación del conjunto $R_\mathcal{P}(I)$, para toda $I \in \mathcal{I}_\mathcal{P}$.

¿Qué es un algoritmo?



Como todo conjunto, $R_\mathcal{P}(I)$ puede definirse:

- Por extensión (i.e., para cada I se almacenan explícitamente sus soluciones)
- Por comprensión (i.e., se define un procedimiento que permita computar para cada I su solución)

¿Qué es un algoritmo?

Una representación por comprensión suele ser lo más eficiente en la práctica, salvo para problemas con conjuntos de instancias y soluciones pequeños

Sin embargo, obtener la solución para una entrada dada puede tomar tiempo

En este curso estaremos interesados en algoritmos que obtienen soluciones de forma eficiente

La importancia del estudio de algoritmos

¿Recuerda el primer algoritmo que conoció/aprendió?

La palabra algoritmo deriva del matemático y erudito árabe Muḥammad ibn Mūsā al-Khwārizmī (Al-Khorezmi, 800 DC)

La *algoritmia* es una de las áreas fundamentales en ciencia de la computación, dedicada al estudio del diseño y análisis de algoritmos

Actualmente un área muy activa en la industria e investigación, aunque tiene su origen mucho antes de la existencia de los actuales computadores...

La importancia del estudio de algoritmos

Los primeros algoritmos registrados son los de multiplicación desarrollados por los egipcios entre los siglos 1700–2000 a.C.

Entre 1600–1800 a.C., los Babilonios desarrollaron algoritmos para resolver problemas como el de multiplicar números naturales

- Traducidos por Donald Knuth en el paper *Ancient Babylonian Algorithms*. Communications of the ACM, Vol. 15 (7), páginas 671–677.

El algoritmo de Euclides para encontrar el mcd, definido en el siglo 300 a.C. y usado hasta el día de hoy, la Criba de Eratóstenes para números primos, del siglo 200 a.C.

Los algoritmos han sido vitales en el desarrollo de las civilizaciones antiguas y continúan siéndolo para la vida moderna

Un ejemplo fundamental: multiplicación de matrices

Sean A y B dos matrices de $n \times n$, se quiere calcular $A \times B$

Principales algoritmos:

- Algoritmo clásico: $\Theta(n^3)$
- Strassen (1969): $\Theta(n^{2,8074})$
- Pan (1978): $O(n^{2,796})$
- Bini, Capovani, Romani (1979): $O(n^{2,780})$
- Schönhage (1981): $O(n^{2,522})$
- Coppersmith, Winograd (1981): $O(n^{2,496})$
- Strassen (1986): $O(n^{2,479})$
- Coppersmith, Winograd (1990): $O(n^{2,3755})$
- Williams (2010): $O(n^{2,3729})$
- Williams, Xu, Xu, Zhou (2023): $O(n^{2,371552})$

Ver la historia completa en `https://en.wikipedia.org/wiki/Computational_complexity_of_matrix_multiplication`

La complejidad de multiplicar matrices

¿Hasta cuándo se podrá seguir mejorando la cantidad de operaciones necesarias para resolver una multiplicación de matrices?

Claramente, no se podrían hacer menos de n^2 operaciones, ya que la matriz resultado tiene n^2 entradas que deben llenarse

En la literatura se conoce como ω al exponente asociado a la cantidad de operaciones $O(n^\omega)$ necesarias para multiplicar dos matrices

¿$\omega = 2$? o ¿$\omega > 2$?

Sólo sabemos que $2 \leq \omega \leq 2{,}371552$

El objetivo de este curso

Introducir técnicas tanto para el diseño como para el análisis de la complejidad computacional de un algoritmo

- Técnicas básicas y avanzadas

Se dará énfasis a:

- El modelo computacional sobre el cual se diseña y analiza un algoritmo
- El uso de ejemplos de distintas áreas para mostrar las potencialidades de las técnicas estudiadas

Algunas consideraciones importantes

Al diseñar un algoritmo debemos considerar el modelo de computación sobre el cual será implementado.

- ¿Qué operaciones podemos realizar en el modelo?

Al analizar la complejidad computacional de un algoritmo también debemos considerar el modelo de computación.

- ¿Qué operaciones consideramos al analizar la complejidad de un algoritmo?

Algunas consideraciones importantes

En general, el análisis de la complejidad de un algoritmo se realiza considerando un tipo particular de entradas.

- El peor caso es muy utilizado, pero también podemos considerar el caso promedio

Al estudiar un problema debemos tener en cuenta qué cotas inferiores se puede demostrar para su complejidad

- Estas cotas inferiores dependen del modelo de computación considerado

Nuestro enfoque para el estudio de algoritmos