

Análisis de algoritmos, complejidad en espacio de memoria y ejecución eficiente de algoritmos

Costes en tiempo y en espacio:

La característica básica que debe tener un algoritmo es que sea correcto, es decir, que produzca el resultado deseado en tiempo finito. Adicionalmente puede interesarnos que sea claro, que este bien estructurado, que sea fácil de usar, que sea fácil de implementar y que sea eficiente.

Entendemos por eficiencia de un algoritmo la cantidad de recursos de cómputo que requiere; es decir, cuál es su tiempo de ejecución y qué cantidad de memoria utiliza.

A la cantidad de tiempo que requiere la ejecución de un cierto algoritmo se le suele llamar coste en tiempo mientras que a la cantidad de memoria que requiere se le suele llamar coste en espacio

Coste en los casos mejor, promedio y peor:

Utilizando la definición anterior caracterizar a la función T puede ser complicado. Por ello se definen tres funciones que dependen exclusivamente del tamaño de las entradas y describen resumidamente las características de la función T .

Definición 2.1 Sea A_n el conjunto de las entradas de tamaño n y $T_n : A_n \rightarrow \mathbb{R}$ la función T restringida a A_n . Los costes en caso mejor, promedio y peor se definen como sigue:

Coste en caso mejor : $T_{\text{mejor}}(n) = \min\{T_n(\alpha) | \alpha \in A_n\}$

Coste en caso promedio : $T_{\text{prom}}(n) = P$

$\alpha \in A_n$

$P r(\alpha) \cdot T_n(\alpha)$, donde $P r(\alpha)$ es

la probabilidad de ocurrencia de la entrada α .

Coste en caso peor : $T_{\text{peor}}(n) = \max\{T_n(\alpha) | \alpha \in A_n\}$

Por lo general en este curso estudiaremos el coste en caso peor de los algoritmos por dos razones:

1. Proporciona garantías sobre el coste del algoritmo ya que en ningún caso excederá el coste en caso peor.

2. Es más fácil de calcular que el coste en el caso promedio.

Una característica esencial del coste de un algoritmo en cualquiera de los casos descritos anteriormente es su tasa de crecimiento o dicho de otra manera su orden de magnitud. La tasa de crecimiento de una función marca una diferen

Análisis de Algoritmos:

Dado que el objeto fundamental de este tema son los algoritmos vamos a empezar por establecer una definición de lo que es un algoritmo: Secuencia finita de instrucciones donde cada una de las cuales tiene un significado preciso y puede ejecutarse con una cantidad finita de recursos computacionales en un tiempo finito.

El análisis de algoritmos es una herramienta para hacer la evaluación del diseño de un algoritmo, permite establecer la calidad de un programa y compararlo con otros que puedan resolver el mismo problema, sin necesidad de desarrollarlos. El análisis de algoritmos estudia, desde un punto de vista teórico, los recursos computacionales que requiere la ejecución de un programa, es decir su eficiencia (tiempo de CPU, uso de memoria, ancho de banda, ...). Además de la eficiencia en el desarrollo de software existen otros factores igualmente relevantes: funcionalidad, corrección, robustez, usabilidad, modularidad, mantenibilidad, fiabilidad, simplicidad y aún el propio costo de programación.

El análisis de algoritmos se basa en:

El análisis de las características estructurales del algoritmo que respalda el programa.

La cantidad de memoria que utiliza para resolver un problema.

La evaluación del diseño de las estructuras de datos del programa, midiendo la eficiencia de los algoritmos para resolver el problema planteado.

Determinar la eficiencia de un algoritmo nos permite establecer lo que es factible en la implementación de una solución de lo que es imposible.

- Tipos de Análisis:

Peor caso (usualmente): $T(n)$ = Tiempo máximo necesario para un problema de tamaño n

Caso medio (a veces): $T(n)$ = Tiempo esperado para un problema cualquiera de

tamaño n

· (Requiere establecer una distribución estadística)

Mejor caso (engañoso): $T(n)$ = Tiempo menor para un problema cualquiera de tamaño n

Referencias:

<https://www.cs.upc.edu/~duch/home/duch/analisis.pdf>

<http://artemisa.unicauca.edu.co/~nediaz/EDDI/cap01.htm>