

Ayudantía 3

Carlos Lagos - carlos.lagosc@usm.cl

Análisis Asintótico

- $f(n) \in O(g(n))$: f crece a lo sumo como una función g
- $f(n) \in o(g(n))$: f crece estrictamente menor a g
- $f(n) \in \Omega(g(n))$: f crece al menos como una función g
- $f(n) \in \omega(g(n))$: f crece estrictamente mayor a g
- $f(n) \in \Theta(g(n))$: f tiene el mismo crecimiento asintótico que g

Análisis a través de límites

- $f(n) \in O(g(n))$:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = k, \quad k \in \mathbb{R}^+ \cup \{0\}, \quad k \neq \infty$$

- $f(n) \in o(g(n))$:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

Análisis a través de límites

- $f(n) \in \Omega(g(n))$:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = k, \quad k \in \mathbb{R}^+$$

- $f(n) \in \omega(g(n))$:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$$

Análisis a través de límites

- $f(n) \in \Theta(g(n))$:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = k, \quad k \in \mathbb{R}^+, \quad k \neq \infty$$

Ejercicios

1. Verdadero o Falso: $n^2 + 3n + 10 \in O(n^2)$
2. Verdadero o Falso: $\sqrt{n} \in \theta(n)$
3. Verdadero o Falso: $a^n \in \theta(b^n)$, con $a \neq b$
4. Verdadero o Falso: $\ln(n) \in \omega(n^{\frac{1}{4}})$
5. Verdadero o Falso: $n\sqrt{n} \in o(n^2)$

Análisis Asintótico

1. Verdadero o Falso: $n^2 + 3n + 10 \in O(n^2)$

$$\lim_{n \rightarrow \infty} \frac{n^2 + 3n + 10}{n^2} = \lim_{n \rightarrow \infty} \left(1 + \frac{3}{n} + \frac{10}{n^2} \right) = 1$$

Es verdadero debido a que tiende a 1 y no a infinito, por ende pertenece a $O(n^2)$.

2. Verdadero o Falso: $\sqrt{n} \in \theta(n)$

$$\lim_{n \rightarrow \infty} \frac{\sqrt{n}}{n} = \lim_{n \rightarrow \infty} \frac{n^{0.5}}{n} = \lim_{n \rightarrow \infty} \frac{1}{n^{0.5}} = 0$$

Es falso debido a que tiende a 0 y para pertenecer a θ tiene que tender a un valor diferente de cero e infinito.

Análisis Asintótico

3. Verdadero o Falso: $a^n \in \theta(b^n)$, con $a \neq b$

$$\lim_{n \rightarrow \infty} \frac{a^n}{b^n} = \lim_{n \rightarrow \infty} \left(\frac{a}{b} \right)^n$$

Caso 1: $a > b$

Para este caso $\frac{a}{b}$ sería estrictamente mayor a 1, por lo que si este número se eleva con un n tendiendo al infinito provoca que $\left(\frac{a}{b}\right)^n$ tienda a infinito. Entonces para este caso no pertenecería a $\theta(b^n)$.

Caso 2: $a < b$

Para este caso $\frac{a}{b}$ sería estrictamente menor a 1, por lo que si este número se eleva con un n tendiendo al infinito provoca que $\left(\frac{a}{b}\right)^n$ tienda a 0. Entonces para este caso no pertenecería a $\theta(b^n)$.

Análisis Asintótico

4. Verdadero o Falso: $\ln(n) \in \omega(n^{\frac{1}{4}})$

Usaremos L'Hôpital:

$$\lim_{n \rightarrow \infty} \frac{\ln(n)}{n^{\frac{1}{4}}} = \lim_{n \rightarrow \infty} \frac{\frac{1}{n}}{\frac{1}{4}n^{-3/4}} = \lim_{n \rightarrow \infty} \frac{4n^{\frac{3}{4}}}{n} = \lim_{n \rightarrow \infty} \frac{4}{n^{1/4}} = 0$$

La aseveración es falsa, ya que para pertenecer a $\omega(n^{\frac{1}{4}})$ tendría que tender a infinito.

5. Verdadero o Falso: $n\sqrt{n} \in o(n^2)$

$$\lim_{n \rightarrow \infty} \frac{n\sqrt{n}}{n^2} = \lim_{n \rightarrow \infty} \frac{\sqrt{n}}{n} = \lim_{n \rightarrow \infty} \frac{n^{\frac{1}{2}}}{n} = \lim_{n \rightarrow \infty} \frac{1}{n^{\frac{1}{2}}} = 0$$

Es verdadero, ya que el límite es diferente de infinito.

Análisis Amortizado

¿Qué es?

En un análisis de este tipo, el tiempo requerido para llevar a cabo una secuencia de operaciones sobre una estructura de datos es promediado sobre todas las operaciones.

¿Cuándo se usa?

Cuando se desea mostrar que el costo promedio de una operación es pequeño, aun cuando una única ejecución de la operación podría ser muy costosa.

Diferencias con Caso Promedio

No utiliza probabilidades, ya que en este caso se garantiza el comportamiento promedio de cada operación en el peor caso.

Método de Agregación

Primero se muestra que, para todo n , una secuencia de n operaciones sobre una estructura de datos toma tiempo de peor caso $T(n)$.

Luego se puede concluir que en el peor caso, el costo amortizado por operación es $T(n)/n$.

Este costo aplica para cualquiera de las operaciones realizadas dentro de la secuencia, independiente del costo o tipo de operación.

Ejercicios

1. Se ejecuta una secuencia de n operaciones sobre una estructura de datos. La i -ésima operación tiene un costo de i si i es un cuadrado perfecto, y de 1 en caso contrario. En este contexto, se busca determinar el costo amortizado por operación.
2. Debes diseñar una estructura llamada `mincola`, que soporte las operaciones de obtener el elemento al frente, sacar el elemento del frente, insertar un elemento al final, y obtener el valor mínimo de la cola. La implementación debe usar únicamente pilas y garantizar un costo amortizado de $O(1)$ por operación.

Dividir y Conquistar

Dividir y conquistar es una técnica de diseño de algoritmos que implica dividir un problema en subproblemas más pequeños y manejables, resolver cada uno de estos subproblemas de forma recursiva, y luego combinar las soluciones de los subproblemas para obtener la solución del problema original.

Notación de Recurrencia

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^d)$$

$$T(n) = \begin{cases} O(n^d) & \text{si } d > \log_b a \\ O(n^d \log n) & \text{si } d = \log_b a \\ O(n^{\log_b a}) & \text{si } d < \log_b a \end{cases}$$

Subarreglo con Suma Máxima

Diseña un algoritmo para encontrar el subarreglo contiguo con la suma máxima en un arreglo de números enteros, utilizando la técnica de dividir y conquistar. El algoritmo debe tener un costo de $O(n \log n)$.

Sucesión de Fibonacci

La sucesión de Fibonacci se puede representar de manera matricial de la siguiente forma:

$$\begin{pmatrix} f_{n-1} & f_n \\ f_n & f_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n$$

Ejemplo: si queremos representar el cálculo de f_5 :

$$\begin{pmatrix} f_4 & f_5 \\ f_5 & f_6 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^5 = \begin{pmatrix} 3 & 5 \\ 5 & 8 \end{pmatrix}$$

Entonces, $f_5 = 5$. Se solicita diseñar un algoritmo de dividir y conquistar que resuelva este problema en $O(\log(n))$.

Desafío ultra difícil

Se te da un array A que consiste de N elementos. El valor de desequilibrio de algún subsegmento de este array es la diferencia entre el elemento máximo y el elemento mínimo de ese segmento. El valor de desequilibrio del array es la suma de los valores de desequilibrio de todos los subsegmentos de este array. Plantea un algoritmo que utilice Dividir y Conquistar y resuelva el problema en $O(N \log(N))$.

Esta en el VJUDGE que está en el repositorio

Demuestre

1. MergeSort usando inducción.
2. SelectionSort usando invariante de ciclo e inducción.

FIN