Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC1253 - Matemáticas Discretas

Ayudantía 11 - Algoritmos y complejidad

08 de noviembre de 2024

Martín Atria, José Thomas Caraball, Caetano Borges

Resumen

1. Algoritmos

Definimos un cerro como una lista consistiendo en una serie estrictamente creciente seguida de una serie estrictamente decreciente. Por ejemplo:

$$[1, 2, 4, 19, 8, 3]$$
$$[-5, 2, 7, 10, 15, 6, 5, 4, 1, 0]$$

son cerros.

1. Escriba un algoritmo que utilice una estrategia de dividir y conquistar que reciba como input un cerro, y entregue como output el valor máximo de este.
2. Calcule la complejidad de su algoritmo.

Intente crear un algoritmo que sea $O(\log(n))$.

2. Complejidad

Sean $f : \mathbb{N} \to \mathbb{R}^+$y $g : \mathbb{N} \to \mathbb{R}^+$dos funciones cualesquiera. Demuestre o entregue un contraejemplo para las siguientes afirmaciones:

1. Si $f(n) \in \Theta(g(n))$ entonces $\text{mín}\,\{f(n), g(n)\} \in \Theta(\,\text{máx}\,\{f(n), g(n)\})$.
2. Si $f(n) \in O(g(n))$ entonces $f(n)^{g(n)} \in O\left(g(n)^{f(n)}\right)$.

3. Complejidad + inducción

Considere la siguiente ecuación de recurrencia:

$$T(n) = \begin{cases} 1 & \text{si } n = 1 \\ 4 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n^2 \log_2(n) & \text{si } n > 1 \end{cases}$$

Demuestre usando inducción que $T(n) \in O\left(n^2 (\log n)^2\right)$. Puede que los siguientes valores le resulten útiles:

$$\log_2(3) \approx 1,6 \quad \log_2(5) \approx 2,3 \quad \log_2(6) \approx 2,6 \quad \log_2(7) \approx 2,8$$

4. Bonus: o chica

Dadas las funciones $f(n) = 2^n$ y $g(n) = n!$, demuestre o entregue un contraejemplo para la siguiente afirmación:

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

Donde $f(n) \in o(g(n))$ si $(\forall c > 0)\,(\exists n_0)\, f(n) < c \cdot g(n), n \geq n_0$.

1. Algoritmos

Definimos un cerro como una lista consistiendo en una serie estrictamente creciente seguida de una serie estrictamente decreciente. Por ejemplo:

$$[1, 2, 4, 19, 8, 3]$$
$$[-5, 2, 7, 10, 15, 6, 5, 4, 1, 0]$$

son cerros.

1. Escriba un algoritmo que utilice una estrategia de dividir y conquistar que reciba como input un cerro, y entregue como output el valor máximo de este.
2. Calcule la complejidad de su algoritmo.

Intente crear un algoritmo que sea $O(\log(n))$.

```
CerroSearch(A = [a_0, ..., a_{n-1}], n)
  if n = 1:
      return A[0]
  m ← ⌊n/2⌋
  if A[m] > A[m+1]:
      return CerroSearch(A[:m], m+1) # ⌈n/2⌉
  else if A[m] < A[m+1]:
      return CerroSearch(A[m+1:], n-m+1) # ⌊n/2⌋
```

b)

$$T(n) = \begin{cases} 1 & , \text{si } n = 1 \\ 3 + T(\lfloor \frac{n}{2} \rfloor) & , \text{si } n > 1 \end{cases}$$

Supongamos que $n = 2^k$, $k \in \mathbb{N}$.

$$\begin{aligned} T(n) = T(2^k) &= T\left(\left\lfloor \frac{2^k}{2} \right\rfloor\right) + 3 = T(2^{k-1}) + 3 \\ &= (T(2^{k-2}) + 3) + 3 \\ 2: \quad &= T(2^{k-2}) + 2 \cdot 3 \\ &= (T(2^{k-3}) + 3) + 2 \cdot 3 \\ 3: \quad &= T(2^{k-3}) + 3 \cdot 3 \end{aligned}$$

$$\vdots$$

$$i: \quad = T(2^{k-i}) + i \cdot 3$$

$$\vdots$$

$$k: \quad = T(2^{k-k}) + k \cdot 3$$

$$= T(1) + 3k$$

$$= 3k + 1$$

Como $n = 2^k \rightarrow k = \log_2(n)$

$$T(n) = 3\log_2(n) + 1 \quad \in \Theta(\log(n) \mid POTENCIA_4)$$

$$\rightarrow \in \Theta(\log(n) \mid POTENCIA_2)$$

Si $f \in \Theta(g \mid POTENCIA_b)$, f y g son asintóticamente no decrecientes y g es b-armónica, entonces $f \in \Theta(g)$.

Asint. no dec: $\exists n_0 \in \mathbb{N}$ tq $\forall n > n_0$, $\overset{g}{f}(n) \leq \overset{g}{f}(n+1)$

$\underset{\hookrightarrow 2}{b}$-armónica: $g(bn) \in \Theta(g(n))$

$g(n) = \log_2(n)$ con $n_0 = 2$ se cumple claramente ya que $\log_2(n)$ es creciente.

$$g(2n) \in \Theta(g(n)) \rightarrow \log_2(2n) \in \Theta(\log_2(n)) = \Theta(\log(n))$$

$$\log_2(2n) = \underbrace{\log_2(2)} + \log_2(n) \leq c\log_2(n)$$

$$1 + \log_2(n) \leq c\log_2(n) \qquad c = 100,\ n_0 = 2$$

BI n_0 : $1 + 1 \leq 100 \cdot 1$ ✓

HI $k \in \{n_0 \ldots n-1\}$

TI . $1 + \log_2(n) = 1 + \log_2(2 \cdot \frac{n}{2})$

$$\xrightarrow{} \frac{n}{2} < n$$

$$= 1 + \log_2(2) + \log_2\left(\frac{n}{2}\right) \leq 2 + 100 \log_2\left(\frac{n}{2}\right)$$

$$= 2 + 100\left(\log_2(n) - \log_2(2)\right)$$

$$= 2 + 100\left(\log_2(n) - 1\right)$$

$$= 100 \log_2(n) - 98 \leq 100 \log_2(n) \quad \square$$

g es 2-armónica

Por Teorema concluimos que $T(n) \in O(\log_2(n)) = O(\log(n))$ $\blacksquare$

2. Complejidad

Sean $f : \mathbb{N} \to \mathbb{R}^+$y $g : \mathbb{N} \to \mathbb{R}^+$dos funciones cualesquiera. Demuestre o entregue un contraejemplo para las siguientes afirmaciones:

1. Si $f(n) \in \Theta(g(n))$ entonces mín $\{f(n), g(n)\} \in \Theta($ máx $\{f(n), g(n)\})$.
2. Si $f(n) \in O(g(n))$ entonces $f(n)^{g(n)} \in O\left(g(n)^{f(n)}\right)$.

1) $h(n) = \min(f(n), g(n))$, $H(n) = \max(f(n), g(n))$

$h(n) \in \Omega(H(n)) \cap O(H(n))$, $\exists c', c'' \in \mathbb{R}^+$ y $\exists n_0$ tq $\forall n \geq n_0$

$$c' H(n) \leq h(n) \leq c'' H(n)$$

$\exists c_1, c_2 \in \mathbb{R}^+$ y $n_0 \in \mathbb{N}$ tq $\forall n \geq n_0$

$$c_1 g(n) \leq f(n) \leq c_2 g(n) \rightarrow \frac{1}{c_2} f(n) \leq g(n)$$

i) $f(n) \leq g(n)$: $h(n) = f(n)$, $H(n) = g(n)$

$$c_1 g(n) \leq f(n) \leq 1 \cdot g(n)$$

$$\min\left(c_1, \frac{1}{c_2}\right) g(n) \leq c_1 g(n) \leq f(n) \leq 1 \cdot g(n)$$

$$\min\left(c_1, \frac{1}{c_2}\right) H(n) \leq h(n) \leq 1 \cdot H(n)$$

ii) $f(n) > g(n)$: $h(n) = g(n)$, $H(n) = f(n)$

$$\frac{1}{c_2} f(n) \leq g(n) \leq f(n)$$

$$\frac{1}{c_2} H(n) \leq h(n) \leq 1 \cdot H(n)$$

$$\min\left(c_1, \frac{1}{c_2}\right) H(n) \leq \frac{1}{c_2} H(n) \leq h(n) \leq 1 \cdot H(n)$$

Tomamos constantes $c' = \min\left(c_1, \frac{1}{c_2}\right)$ y $c'' = 1$, $n_0 = n_0$

$c' H(n) \leq h(n) \leq c'' H(n)$

2. Si $f(n) \in O(g(n))$ entonces $f(n)^{g(n)} \in O\left(g(n)^{f(n)}\right)$.

$f(n) = 2 \qquad g(n) = n \qquad 1 \in O(n)$

$$\left.\begin{aligned} f(n)^{g(n)} &= 2^n \\ g(n)^{f(n)} &= n^2 \end{aligned}\right\} \quad 2^n \notin O(n^2)$$

X

3. Complejidad + inducción

Considere la siguiente ecuación de recurrencia:

$$T(n) = \begin{cases} 1 & \text{si } n = 1 \\ 4 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n^2 \log_2(n) & \text{si } n > 1 \end{cases}$$

Demuestre usando inducción que $T(n) \in O\left(n^2 (\log n)^2\right)$. Puede que los siguientes valores le resulten útiles:

$$\log_2(3) \approx 1,6 \quad \log_2(5) \approx 2,3 \quad \log_2(6) \approx 2,6 \quad \log_2(7) \approx 2,8$$

$4T(\lfloor \frac{n}{2} \rfloor) + n^2 \log_2(n) \leq c\, n^2 \log_2(n)^2 \quad \forall n \geq n_0 \in \mathbb{N}.$

$c = 10$

$T(1) = 1 \qquad$ LD: $0 \qquad$ X

$T(2) = 4 \cdot T(1) + 4 \log_2(2) \qquad$ LD: $10 \cdot 4 \cdot \log_2(2)^1$

$\quad = 4 + 4 \qquad \leq \qquad 10 \cdot 4 \quad \checkmark$

$T(3) = 4 \cdot T(1) + 9 \log_2(3)$

$\quad = 4 + 9 \cdot 1,6 = 18,4$

LD: $10 \cdot 9 \cdot 1,6 \geq 18,4 \quad \checkmark$

HI: Supongamos que $T(k) \leq 10\, k^2 \log_2(k)^2 \quad \forall k \in \{2, \ldots, n-1\}$

TI: PD: $T(n) \leq 10 n^2 \log_2(n)^2$

$T(n) = 4\,T(\lfloor \frac{n}{2} \rfloor) + n^2 \log_2(n) \qquad$ / HI: $\lfloor \frac{n}{2} \rfloor < n \quad \lfloor \frac{n}{2} \rfloor \geq 2$

$\leq 4 \cdot 10 \lfloor \frac{n}{2} \rfloor^2 \log_2 (\lfloor \frac{n}{2} \rfloor)^2 + n^2 \log_2(n)$

$\leq 40 (\frac{n}{2})^2 \log_2 (\frac{n}{2})^2 + n^2 \log_2(n)$

$$= 10n^2\left(\log_2\left(\frac{n}{2}\right)^2 + \log_2(n)\right)$$

$$= 10n^2\left([\log_2(n) - 1]^2 + \log_2(n)\right)$$

$$= 10n^2\left(\log_2(n)^2 - 2\log_2(n) + 1 + \log_2(n)\right)$$

$$= 10n^2\left(\log_2(n)^2 + \underbrace{1 - \log_2(n)}\right)$$

$$n \geq 4 \rightarrow \log_2(n) \geq 1 \rightarrow 1 - \log_2(n) < 0$$

$$\rightarrow \leq 10n^2 \log_2(n)^2 \quad \square$$

Concluímos que $T(n) \in O(n^2 \log(n)^2)$. $\square$

4. Bonus: o chica

Dadas las funciones $f(n) = 2^n$ y $g(n) = n!$, demuestre o entregue un contraejemplo para la siguiente afirmación:

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

Donde $f(n) \in o(g(n))$ si $(\forall c > 0)\ (\exists n_0)\ f(n) < c \cdot g(n), n \geq n_0$.

$$f(n)\in o(g(n)) \Leftrightarrow \underline{(\forall c \in \mathbb{R}^+)}(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)(f(n) \leq c\,g(n))$$

$$2^n \leq cn! \to \frac{2^n}{cn!} \leq 1 \to \frac{1}{c}\,\frac{2^n}{n!}$$

$$= \frac{1}{c}\cdot\frac{2}{n}\underbrace{\left(\frac{2}{n-1}\cdot\frac{2}{n-2}\cdots\frac{2}{3}\right)}_{\prod < 1}\frac{2}{2}\cdot\frac{2}{1}$$

$$\leq \frac{1}{c}\cdot\frac{2}{n}\cdot\frac{2}{1} = \frac{4}{cn} \leq 1$$

$\to \forall n \geq n_0$ con n_0 tq $\frac{4}{c} \leq n_0$

Si tomamos $n_0 = \left\lceil \frac{4}{c} \right\rceil + 1$

Concluimos que $2^n \in o(n!)$ □.