Examen 1

Análisis y Diseño de Algoritmos

22 de Abril del 2022

Ejercicio 1 (3 ptos). Pruebe que ✓

$$n^2 - 100n - 200\sqrt{n} + 50\sqrt[3]{n} = \Theta(n^2).$$

Ejercicio 2 (5 ptos). Considere la siguiente recurrencia: ✓

$$T(n) = \begin{cases} 1 & n = 1 \\ 4T(\lfloor n/4 \rfloor) + n^2 & \text{caso contrário} \end{cases}$$

- Resuelva la recurrencia **de manera explícita**, usando los métodos vistos en clase. ✓ Debe dar *la solución exacta* cuando n es potencia de 4. Cuando no lo es, debe acotar adecuadamente. Puede suponer que es conocido que $T(n)$ es una función creciente. ✓
- Demuestre *por inducción* que $T(n)$ es $\Omega(n \lg n)$ (no puede utilizar el resultado anterior). ✓

Ejercicio 3 (4 ptos). ✓ Un segmento de un vector $A[1..n]$ es cualquier vector de la forma $A[p..q]$, con $1 \leq p \leq q \leq n$. El tamaño de dicho segmento es $q - p + 1$. El segmento es *creciente* si $A[p] \leq \cdots \leq A[q]$. Problema: dado un vector $A[1..n]$ devuelva el tamaño máximo de un segmento creciente en A.

- Escriba un algoritmo de división y conquista con complejidad $\Theta(n \lg n)$. Puede asumir que n es potencia de 2.
- Escriba la recurrencia que define el peor caso en el tiempo de ejecución del algoritmo. Puede asumir que todas las constantes asociadas valen 1.
- Resuelva la recurrencia anterior usando teorema maestro

Ejercicio 4 (3 ptos). Pruebe *por inducción* que la altura de un nodo i de un heap $A[1..n]$ es menor o igual que $\lg \frac{n}{i}$. Puede saber que es conocido cuanto es el valor de la altura de un heap con n nodos (no necesita probar este hecho).

Ejercicio 5 (3 ptos). Considere el siguiente algoritmo que recibe un arreglo $A[1..n]$ de números.

ALGO (A)

for $j \leftarrow n$ down to 2
  $i \leftarrow j$
  while $i \geq 2$ and $A[\lfloor i/2 \rfloor] < A[i]$
    $A[\lfloor i/2 \rfloor] \leftrightarrow A[i]$
    $i \leftarrow \lfloor i/2 \rfloor$

El algoritmo promete reorganizar A como Max-heap.

- ¿El algoritmo hace lo prometido? Justifique.
- En caso negativo, indique como modificar *una sola línea del algoritmo* para que lo haga.

Ejercicio 6 (2 ptos). Esta parte es electiva. *Responda solo una de las siguientes preguntas (si responde más de una no se le considerará puntaje).*

- En el algoritmo de división y conquista para calcular las inversiones. Demuestre que, luego de terminar las llamadas recursivas, si (i, j) es una inversión con $i \leq \lfloor (p+r)/2 \rfloor < j$, entonces (i', j') es también una inversión si $i < i' \leq \lfloor (p+r)/2 \rfloor$ y $\lfloor (p+r)/2 \rfloor \leq j' < j$.
- Indique cuales son las tres llamadas recursivas en el algoritmo de Karatsuba y explique por qué es suficiente hacer estas únicas tres llamadas.

CÓDIGO

2 0 2 0 1 0 3 1 1

NOMBRE DEL ALUMNO

MATOS	CANGALAYA	JEREMY JEFFREY
APELLIDO PATERNO	APELLIDO MATERNO	NOMBRES

CARRERA

CS

SECCIÓN

1

FIRMA

PRÁCTICA CALIFICADA DEL CURSO:

Nº 1

NOTA

~~10~~ 17 DIECISIETE

NOMBRE DEL PROFESOR

Juan [illegible]

FIRMA

IMPORTANTE

- El alumno deberá completar los datos indicados en la carátula antes de iniciar la práctica.
- La práctica deberá ser desarrollada únicamente con lapicero negro o azul.
- El trabajo en limpio se desarrollará en la cara derecha de cada hoja, el lado izquierdo podrá ser usado como borrador.
- Se considerará importante para la calificación el orden, la ortografía y limpieza de la práctica.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

$$\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$$

$$\frac{\left(\frac{1}{4}\right)^{\log_4 n - 1 + 1} - 1}{\frac{1}{4} - 1}$$

$$\frac{\frac{1}{n}-1}{\frac{-3}{4}} = \frac{\frac{1-n}{n}}{\frac{-3}{4}} = \frac{4-4n}{-3n} = \frac{4n-4}{3\cancel{n}} \cdot n^{\cancel{2}}$$

21
16
126
21
336

$n=4$

$T(4)=4T(1)+16$

20

$n=16$

$4\underbrace{T(4)}_{20} + 16^2 =$

$80 + 256$

336

$n + n^2 \left(\frac{4-4n}{-3n}\right)$

$16 + 256\left(\frac{4-64}{-3\cdot 16}\right)$

$16 + \cancel{256}^{64}\left(\frac{-\cancel{60}^{5}}{-\cancel{48}_{\cancel{4}\,1}}\right)$

$16 + 320$

336

$n + \frac{4n^2-4n}{3} = n + n\left(\frac{4n-4}{3}\right)$

$4 + 16 = 20$

$n\left(\frac{4n-4}{3}+1\right)$

$16 + 4. \quad -64$

$n\left(\frac{4n-1}{3}\right)$

16

$16(21)$

$\underbrace{x_l \quad x_r}_{}$ X $\quad$ $\underbrace{y_l \quad y_r}_{}$ Y

$(x_l + x_r)(y_l + y_r) =$

$x_l y_l + x_r y_r + x_r y_l + x_l y_r$

$\frac{\cancel{n}}{4} \le \frac{n^{\cancel{2}}}{2}$

$\frac{1}{2} \le n$

$-2n \le \frac{1}{2} n \lg(n)$

$-4\cancel{n} \le \cancel{n} \lg(n)$

$n \ge 1$

$T(n) \ge n \lg n . c \ge 0$

$n \ge 8$

$T(n) = 4T(n/4) + n^2$

$\ge \cancel{4} c \frac{n}{\cancel{4}} \lg\left(\frac{n}{4}\right) + n^2$

$\ge n(\lg(n) - \lg(4)) + n^2$

$c n \lg(n) - c2n + n^2 \ge n\lg(n) \underbrace{-2n}_{\le \frac{1}{2} n \lg(n)}$

$= n(\lg(n) - 2)$

$n^2 - c2n \ge 0$

$n^2 - n \ge 0$

$-2\cancel{n} + n^{\cancel{2}} \le \frac{1}{2} \cancel{n} \lg n$

$n - 2 \le \frac{1}{2} \lg(n)$

$2n - 4 \le \lg(n)$

E-2

$$T(n) = \begin{cases} 1 \\ 4T(\lfloor n/4 \rfloor) + n^2 \end{cases}$$

$n = 4^K$

$$T(n) = 4T(n/4) + n^2$$

$n = 4^i$

$\lg_4 n = i$

$$= 4\left(4T(n/4^2) + \left(\frac{n}{4}\right)^2\right) + n^2$$

$$= 4^2 T(n/4^2) + \frac{n^2}{4} + n^2$$

$$= 4^2\left(4T(n/4^3) + \left(\frac{n}{4^2}\right)^2\right) + \frac{n^2}{4} + n^2$$

$$= 4^3 T(n/4^3) + \frac{n^2}{4^2} + \frac{n^2}{4^1} + \frac{n^2}{4^0}$$

$$\vdots$$

$$= 4^i T(n/4^i) + n^2\left(\frac{1}{4^0} + \frac{1}{4^1} + \dots + \frac{1}{4^{i-1}}\right)$$

$i = \lg_4 n$

$$\vdots$$

$$4^{\lg_4 n} T(1) + n^2 \underbrace{\sum_{K=0}^{\lg_4 n - 1} \frac{1}{4^K}}_{\text{geometrica}} = n + n^2\left(\frac{\left(\frac{1}{4}\right)^{\log_4 n} - 1}{1/4 - 1}\right)$$

$$= n + n^2\left(\frac{4 - 4n}{-3n}\right)$$

$$T(n) = n\left(\frac{4n-1}{3}\right) \blacksquare$$

2

$4^K \leq n < 4^{K+1}$

- $T(4^K) = 4^K\left(\frac{4^{K+1}-1}{3}\right) = \frac{4^{2K+1} - 4^K}{3} = \frac{(4^K)^2 \cdot 4 - 4^K}{3} \leq \frac{4n^2 - n}{3} \geq \frac{4n^2 - n}{4} = n^2 - \underbrace{\frac{n}{4}}_{\leq 1/2 n^2}$

$$T(4^K) \geq n^2 - \frac{n}{4} \geq n^2 \cdot \frac{1}{2} \quad \forall n \geq 1$$

- $T(4^{K+1}) = 4^{K+1}\left(\frac{4^{K+2} - 1}{3}\right) \leq 4^{K+1}\left(\frac{4^{K+2}}{3}\right) \leq \frac{4^{2K+3}}{3} = \frac{(4^K)^2 \cdot 4^3}{3} \leq \frac{64}{3} n^2 \quad \forall n \geq 1$

Entonces:

$$0 \leq \frac{1}{2} n^2 \leq \underbrace{n\left(\frac{4n-1}{3}\right) \leq \frac{64}{3} n^2}_{T(n) = \Theta(n^2)} \quad \forall n \geq 1$$

0

Probar $T(n) = \Omega(n \lg n)$ $\qquad c = \frac{1}{2}$

Caso base: $T(1) = 1 \geq 1 \lg(1) \cdot \frac{1}{2} = 0$

Paso inductivo: Suponga que $T(\lfloor n/4 \rfloor) \geq \frac{1}{2} \lfloor \frac{n}{4} \rfloor \lg \lfloor \frac{n}{4} \rfloor$

Luego: $T(n) = 4T(\lfloor n/4 \rfloor) + n^2$

$$\geq \cancel{4} \cdot \frac{1}{2} \frac{n}{\cancel{4}} \lg(\lfloor n/4 \rfloor) + n^2 \quad \text{por (H.I)}$$

$$\geq \frac{n}{2}\left(\lg(n) - \underbrace{\lg(4)}_{2}\right) + n^2$$

$$\geq \frac{n}{2}\lg(n) \underbrace{- n + n^2}_{\geq 0 \ \forall n \geq 1} \geq \frac{n}{2}\lg(n) \geq 0 \quad \forall n \geq 1$$

1,5

$\therefore \quad 0 \leq \frac{1}{2} n \lg(n) \leq T(n) \quad \forall n \geq 1 \qquad T(n) = \Omega(n \lg n) \blacksquare$

E-6

- Karatsuba sean 2 números X, Y

$$\left.\begin{array}{l} X = X_l \cdot 10^{n/2} + X_r \\ Y = Y_l \cdot 10^{n/2} + Y_r \end{array}\right\} XY = \underbrace{(X_l \cdot Y_l)}_{1} 10^n + (\underbrace{X_l Y_r}_{2} + \underbrace{Y_l X_r}_{3}) 10^{n/2} + \underbrace{X_r Y_r}_{4}$$

Este algoritmo haría 4 llamadas recursivas:

Karatsuba (X, Y)
  p = Karatsuba(X_l, Y_l)
  q = Karatsuba(X_l, Y_r)
  r = Karatsuba(X_r, Y_l)
  s = Karatsuba(X_r, Y_r)
  return $(p)10^n + (q+r)\,10^{n/2} + s$

$T(n) = 4T(n/2)$

Sin embargo:

$$(X_l + X_r)(Y_l + Y_r) = \underbrace{X_l Y_l + X_r Y_r}_{\text{son 2 conocidas}} + X_l Y_r + X_r Y_l$$

$$X_l Y_r + X_r Y_l = \underbrace{(X_l + X_r)(Y_l + Y_r)}_{1} - \underbrace{X_l Y_l}_{2} - \underbrace{X_r Y_r}_{3}$$

Lo anterior concluye que son necesarias solo 3 llamadas.

Resultando en: Karatsuba (X, Y)
  p = Karatsuba(X_l, Y_l)
  q = Karatsuba(X_r, Y_r)
  s = Karatsuba($X_l + X_r$, $Y_l + Y_r$)
  return $p\,10^n + (s - q - p)\,10^{n/2} + s$

$T(n) = 3T(n/2) + n^2$

$T(n) = \Theta(n^{\lg 3})$

B 2

$\underbrace{n^2 - 100n}_{< n^2/4} - \underbrace{200\sqrt{n}}_{< n^2/4} + \underbrace{50\sqrt[3]{n}}_{n^2/4}$

$1\frac{n^2}{4} \leq f(n) \leq$

$100n < \frac{n^2}{4}$

$400 < n$

$200\sqrt{n} < \frac{n^2}{4}$

$800\sqrt{n} < n^2$

$\sqrt{n} < \frac{n^2}{800}$

$800^2 n < n^4$

$800^2 < n^3$

$\sqrt[3]{800^2} < n$

$n^2 + 50\sqrt[3]{n} \leq 2n^2$

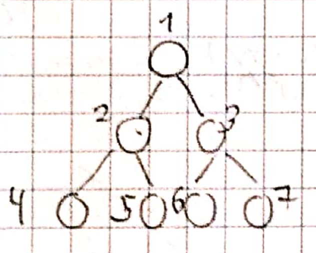
$50\sqrt[3]{n} \leq \left(\frac{n^2}{50}\right)^3$

$(\sqrt[3]{n})^3 \leq \left(\frac{n^2}{50}\right)^3$

$n \leq \frac{n^6}{50^3}$

$50^3 \leq n^5$

$\sqrt[5]{50^3} \leq n$



$n = \lfloor \lg n \rfloor$

$\lg(n) - \lg(i)$

E-1

$$n^2 - 100n - 200\sqrt{n} + 50\sqrt[3]{n} = \Theta(n^2)$$

Note que para $n \geq \max\{400, \sqrt[3]{800^2}\}$ se tiene que

con $n \geq 400$:

$$n \geq 400$$
$$n^2 \geq 400n$$
$$\frac{n^2}{4} \geq 100n$$

con $n \geq \sqrt[3]{800^2}$:

$$\sqrt[3]{800^2} \leq n$$
$$800^2 \leq n^3$$
$$800^2 n \leq n^4$$
$$n \leq \frac{n^4}{800^2}$$
$$\sqrt{n} \leq \frac{n^2}{800}$$
$$800\sqrt{n} \leq n^2$$
$$200\sqrt{n} \leq n^2/4$$

3

Luego

$$n^2 - 100n - 200n + 50\sqrt[3]{n} \geq n^2 - 100n - 200n \geq n^2 - 2\frac{n^2}{4} = \frac{1}{2}n^2$$

Con $n \geq \sqrt[5]{50^3}$:

$$\sqrt[5]{50^3} \leq n$$
$$50^3 \leq n^5$$
$$n \cdot 50^3 \leq n^6$$
$$n \leq \frac{n^6}{50^3}$$
$$\sqrt[3]{n} \leq \frac{n^2}{50}$$
$$50\sqrt[3]{n} \leq n^2$$

Luego

$$n^2 - 100n - 200n + 50\sqrt[3]{n} \leq n^2 + 50\sqrt[3]{n} \leq n^2 + n^2 = 2n^2$$

Entonces:

$$0 \leq \frac{1}{2}n^2 \leq n^2 - 100n - 200\sqrt{n} + 50\sqrt[3]{n} \leq 2n^2 \qquad \forall n \geq \max\{400, \sqrt[3]{800^2}, \sqrt[5]{50^3}\}$$

$\blacksquare$

E-4

$\lg(n) - \lg(i) \rightarrow$ profundidad del nodo

↓

altura del arbol = profundidad

¿Cual es el parametro de la inducción, no i?

Caso base: 1 nodo $\quad \lg(1/1) = 0$

Paso inductivo: Suponga que altura del nodo $n/2$ (ultimo con hijos) es

$$\lg(n/n/2) = \lg(2n/n) = \lg(2) = 1$$

Luego si tomamos un hijo del nodo $n/2$ (ultimo con hijos) esta seria una hoja con altura 0

altura del padre -1 = altura del hijo $\blacksquare$

0

E-3

max-subsegment (A, l, r) basado en el maxsubarray problem

if l<r: mid = l + (r-l)/2 $\Theta(1)$

S_1 = max-subsegment (l, mid) $T(n/2)$

S_2 = max-subsegment (mid+1, r) $T(n/2)$

S_3 = max-crossing-subsegment (A, l, mid, r) $\Theta(n)$

return max (S_1, S_2, S_3) $\Theta(1)$

max-crossing subsegment (A, l, mid, r)

i = mid

while i ≥ l and A [i-1] ≤ A[i]:

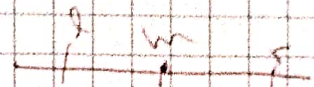
i --

j = mid

while j < r and A[j] < A[j+1]:

j ++

return j - i + 1



Recurrencia : $2T(n/2) + \Theta(n)^1 = \Theta(n \lg n)$

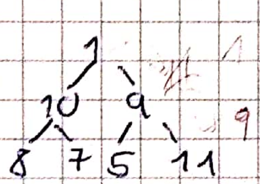
$\lg_2 2/\lg_2 2 = 1$ → teorema maestro

4

E-5

• No, porque hacer intercambios hacia arriba no garantiza que los subárboles se mantengan como máx heap.

Contraejemplo : correr en heap



• Cambio

línea 1: ir de j ← 2 up to n ya que después de cada iteración garantiza que lo de arriba es un max-heap

Algo (A)

for j ← 2 to n

i ← j

while i ≥ 2 and A[⌊i/2⌋] < A[i]

A[⌊i/2⌋] ↔ A[i]

i ← ⌊i/2⌋

3