

Análisis de Algoritmos

Primer examen parcial

a entregar el lunes 12 de octubre del 2020, 16h00

Instrucciones:

1. Resuelve este examen de forma individual.
 2. A enviar en formato *.pdf
 3. Despues de la entrega del examen tendremos una pequeña entrevista individual. Prever tiempo entre 16h y 18h.
-

Opción múltiple (20pts)

1. ¿Cuál es el caso que es verdadero trivialmente al realizar una prueba por inducción? (2pts)
A. la base
B. la hipótesis inductiva
C. la invariante al ciclo
D. el lema
E. ninguno de los anteriores
2. Un algoritmo recursivo trabaja resolviendo dos problemas de la mitad del tamaño recursivamente con un *overhead* (tiempo que toma el procesamiento fuera de la llamada recursiva) de tiempo de ejecución lineal. El tiempo de ejecución está descrito de manera más precisa por: (2pts)
A. $O(\log n)$ B. $O(n)$ **C. $O(n \log n)$** D. $O(n^2)$ E. ninguno de los anteriores.
3. ¿Cuál de las siguientes funciones crece más rápido? (3pts)
A. $n \log n$ **B. 2^n** C. $\log n$ D. n^2 E. n^{20}
4. ¿Cuál de las siguientes funciones crece más rápido? (3pts)
A. $n + \log n$ **B. $n \log n$** C. $n - \log n$ D. n E. dos o más funciones crecen a la misma velocidad.
5. Junto a cada uno de los cinco siguientes términos, escribe la letra de su definición apropiada. (Nota que hay más definiciones que términos. Lee las definiciones con cuidado. (10pts)

$$T(y) = T\left(\frac{y}{2}\right) + O(C_y)$$

$\Theta(g(n))$ **B**

$\Omega(g(n))$ **D**

$w(g(n))$ **H**

$O(g(n))$ **F**

$o(g(n))$ **G**

Definiciones:

- A. $\{f(n)\}$: para cualquier par de constantes c_1 y c_2 positivas existe una constante $n_0 > 0$ tal que $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ para todo $n \geq n_0$.
- B. $\{f(n)\}$: existen constantes positivas c_1, c_2 y n_0 tal que $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ para toda $n \geq n_0$.
- C. $\{f(n)\}$: existen constantes positivas c y n_0 tal que $0 \leq g(n) \leq c f(n)$ para toda $n \geq n_0$.
- D. $\{f(n)\}$: existen constantes positivas c y n_0 tal que $0 \leq c g(n) \leq f(n)$ para toda $n \geq n_0$.
- E. $\{f(n)\}$: para cualquier constante positiva $c > 0$ existe una constante $n_0 > 0$ tal que $0 \leq c g(n) \leq f(n)$ para toda $n \geq n_0$.
- F. $\{f(n)\}$: existen constantes positivas c y n_0 tal que $0 \leq f(n) \leq c g(n)$ para toda $n \geq n_0$.
- G. $\{f(n)\}$: para cualquier constante positiva $c > 0$ existe una constante $n_0 > 0$ tal que $0 \leq f(n) < c g(n)$ para toda $n \geq n_0$.
- H. Ninguno de los anteriores

Stable Marriage Problem (25pts)

6. Ejecuta el algoritmo de Gale-Shapley en el siguiente conjunto de hombres y mujeres. Considera que los hombres proponen y las mujeres aceptan. Los nombres están abreviados con su primera letra.

Hombres	Preferencias	Mujeres	Preferencias
Adrián	$S > T > U > V$	Susana	$D > A > B > C$
Bruno	$S > V > U > T$	Tania	$A > D > C > B$
Carlos	$S > U > T > V$	Ursula	$C > D > B > A$
David	$U > V > T > S$	Virginia	$A > C > D > B$

7. Determina una lista de cuatro hombres y cuatro mujeres (cada uno ordenando a las cuatro personas del sexo opuesto por preferencia) donde ninguno obtenga su primera opción, sin importar si son los hombres o las mujeres quienes proponen.

8. Prueba que si los hombres proponen, entonces a lo más, uno de ellos tiene su última opción (suponiendo que todos ordenaron en orden de preferencia al sexo opuesto).

Recurrencias (30pts)

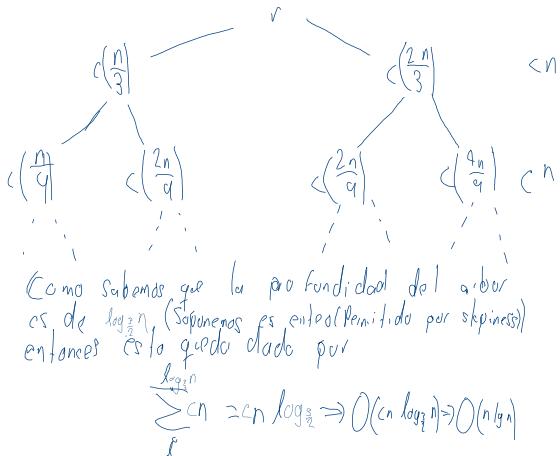
9. Prueba por el método de sustitución que la cota inferior de la siguiente recurrencia es $T(n) = n^2 \lg(n)$:

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

10. Usa un árbol de recusión para determinar la función de crecimiento asintótico (apretada) para la recurrencia $T(n) = T(n/3) + T(2n/3) + O(n)$. Utiliza el método de sustitución para verificar tu respuesta. (15pts)

Page 2

$$\textcircled{O} \quad T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n)$$



Ahora probaremos esto por sustitución. De manera que siendo $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n)$ y la aproximación $O(\cdot) = n \lg n$, tenemos

$$\begin{aligned} T(n) &= T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n) \\ &\leq \frac{1}{3}n \lg\left(\frac{n}{3}\right) + \frac{2}{3}n \lg\left(\frac{2n}{3}\right) + cn \\ &\leq \frac{1}{3}\left(n \lg\left(\frac{n}{3}\right) + \frac{2n}{3} \lg\left(\frac{2n}{3}\right)\right) + cn \\ &= \frac{1}{3}\left(n \lg n - \frac{1}{3}n \lg 3 + \frac{2n}{3} \lg 2 + \frac{2n}{3} \lg 3 - \frac{2n}{3} \lg 2\right) + cn \\ &= \frac{1}{3}\left(n \lg n - \frac{1}{3}n \lg 3 + \frac{2n}{3} \lg 3 - \frac{2n}{3} \lg 2\right) + cn \\ &= \frac{1}{3}\left(n \lg n - \frac{1}{3}n \lg 3 - \frac{2n}{3} \lg 2\right) + cn \\ &= \frac{1}{3}n \lg n - \frac{1}{9}n \lg 3 - \frac{2}{3}n \lg 2 + cn \\ &\leq \frac{1}{3}n \lg n \end{aligned}$$

Por lo que, para un $\delta \geq \frac{c}{19/3 - 2/3}$ se cumple el límite y podemos afirmar que

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n) \in O(n \lg n)$$

⑥ Sabemos que el algoritmo es ya definido en el siguiente bucle

1. Cada chico invita a su primera opción
2. Cada chica acepta la mejor
3. Cada chico rechazado regresa a 1.

Con los datos dados se dan las siguientes iteraciones

Iter	Proposiciones	Parejas formadas
1	$A \rightarrow S, B \rightarrow S, C \rightarrow S, D \rightarrow U$	$(A, S), (D, U)$
2	$B \rightarrow V, C \rightarrow U$	$(A, S), (B, V)$
3	$D \rightarrow V$	$(A, S), (C, U), (D, V)$
4	$B \rightarrow U$	$(A, S), (C, U), (D, V)$
5	$B \rightarrow T$	$(A, S), (B, T), (C, U), (D, V)$

Y como verificamos que el empalmamiento de S es estable, ya tenemos

A	V	U	S	T
B	U	V	S	T
<	S	T	UV	
D	T	S	UV	

S	A	B	C
T	A	B	C
U	C	A	B
V	C	B	A

Al finalizar algoritmo estaron con primera, pero por condición de estabilidad eso cambia

$$\begin{aligned} \textcircled{O} \quad &Vamos a demostrar que \\ &T(n) \in O(n^2 \lg n) \\ &\text{de esto (por sustitución) quedan} \\ &T(n) = 4T\left(\frac{n}{2}\right) + n^2 \\ &\leq 4\left(\left(\frac{n}{2}\right)^2 + \lg\left(\frac{n}{2}\right) + O\left(\frac{n}{2}\right)^2\right) \\ &\leq 4\left(\frac{n^2}{4} + (\lg n - \lg 2) + O(n^2)\right) \\ &\leq n^2 \cdot (\lg n - 1) + O(n^2) \\ &= n^2 \lg n - n^2 + O(n^2) \\ &\leq n^2 \lg n \end{aligned}$$

11. Resuelve las siguientes recurrencias utilizando cualquier método para encontrar su función (apretada) de crecimiento asintótico con notación Θ . Justifica tus respuestas. (20pts)

$$(a) T(n) = 2T(n/3) + n \log n$$

$$(b) T(n) = 7T(n/2) + n^3$$

Mergesort (10pts)

El algoritmo de Mergesort visto en clase funciona como sigue para una lista no ordenada U de n números:

- Si $n = 1$, regresa U (ya está ordenada).
- Divide U en dos listas U_1 y U_2 (aproximadamente del mismo tamaño.)
- Ordena recursivamente: $S_1 = \text{Mergesort}(U_1)$ y $S_2 = \text{Mergesort}(U_2)$
- Regresa la salida: $S = \text{Merge}(S_1, S_2)$.

12. Da la relación de recurrencia que describe el tiempo de ejecución de Mergesort.

13. Da la cota al tiempo de ejecución del algoritmo Mergesort utilizando la notación O . Prueba el resultado de esta cota.

Análisis Probabilístico (15pts)

14. En el problema de contratación de asistente HIRE-ASSISTANT visto en clase; suponiendo que los candidatos se presentan en orden aleatorio, ¿Cuál es la probabilidad de que contrates exactamente una vez? ¿Cuál es la probabilidad de que contrates exactamente n veces?

15. Utiliza variables indicadoras para calcular el valor esperado de la suma de n dados.

Page 3

$$11.a) T(n) = 2T\left(\frac{n}{3}\right) + n \log n$$

Usando el teorema maestro tenemos que

$$a=2, b=3, l=\log_b a = \log_3 2 \approx 0.667$$

Probando el caso 3. Primero observemos que

$$n \log n \in \Omega(n^l)$$

y que con $\epsilon \approx 0.4$, tenemos que

$$\Omega(p^{l+\epsilon}) \sim \Omega(n^l)$$

por lo que cumple la primera condición

Respecto a si $3 < 1 + \epsilon$ o $\omega(p) \leq \omega(n)$

tendremos que

$$2T\left(\frac{n}{3}\right) \leq C f(n)$$

$$2\left(\frac{n}{3} \log \frac{n}{3}\right) \leq C n \log n$$

$$\frac{2n}{3} (\log n - \log 3) \leq C n \log n$$

$$\frac{2n}{3} (\log n - \log 3) \leq C n$$

$$\frac{2n}{3} (1 - \log_3 2) \leq C n$$

$$\frac{2n}{3} (1 - \log_3 2) \leq C$$

$$\frac{2n}{3} - \frac{2 \log_3 2}{3} \leq 1 \leq C$$

Por lo que queda, por el teorema maestro,

$$T(n) \in O(n \log n)$$

$$11.b) T(n) = 7T\left(\frac{n}{2}\right) + n^3$$

Igual usamos el teorema maestro, de donde tenemos que

$$a=7, b=2, l=\log_b a = \log_2 7 \approx 3$$

Probamos el caso 3, donde tenemos que

$$n^3 \in \Omega(n^3)$$

y como tenemos que $l < 3$ no existe un $\epsilon > 0$ tq $l + \epsilon = 3$, por lo que cumple la primera condición. Respecto a la segunda se da que

$$c_1 f\left(\frac{n}{2}\right) \leq C f(n)$$

$$2f\left(\frac{n}{2}\right) \leq C f(n)$$

$$2 \cdot \frac{n^3}{8} \leq C n^3$$

$$\frac{3}{4} n^3 \leq C n^3$$

entonces también cumple la segunda condición. Por lo que, por el teorema maestro

$$T(n) \in O(n^3)$$

$$12) \text{Suponemos Merge toma } \Theta(n)$$

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

$$13) \text{Usando el teorema maestro tenemos}$$

$$a=2, b=2, l=\log_b a = 1$$

probamos el caso 2. Donde

$$\Theta(n) \in O(n)$$

y con $l=1 \Rightarrow n^1 \in O(n)$, entonces se cumple la condición y, por el teorema maestro, se da que

$$T(n) \leq C(n^1 \log n)$$

$$\leq C(n \log n)$$

Como nos piden O , sabemos que $g \in O(n) \Leftrightarrow g \in O(1) \cap g \in \Omega(n)$

por lo que

$$T \in O(n \log n) \Rightarrow T \in O(n \log n)$$

$$14) \text{Tendremos que la probabilidad}$$

de contratar n veces es igual a la probabilidad de que vengan ordenados del peor al mejor, la cual es

$$\frac{1}{n!} = \frac{1}{n!}$$