Examen

Alumno: Axel Daniel Malváez Flores
Matemáticas Discretas

1. Resuelve las siguientes recursiones:

- $T(n) = 7T\left(\frac{n}{2}\right) + 4^n$

Como $f(n) = 4^n$ notamos que podemos aplicar el tercer caso del teorema maestro.

Es claro que $f(n) = \Omega\left(n^{\log_2 7}\right)$ pues una función cuadrática acota por abajo a una función exponencial. Ahora verificamos que se satisface la condición de regularidad:

$$af\left(\frac{n}{b}\right) < cf(n)$$

$$7f\left(\frac{n}{2}\right) < c\,4^n$$

$$7 \cdot 4^{\frac{n}{2}} < c\,4^n$$

$$\frac{7 \cdot 4^{\frac{n}{2}}}{4^n} < c$$

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

$$\frac{7}{4^{\frac{n}{2}}} < c$$

Cuando n es suficientemente grande $\frac{7}{4^{\frac{n}{2}}} \xrightarrow{\text{tiende}} 0$ entonces la "c" queda acotada como $0 < c < 1$, sea $c = 0.5$

Por tanto se satisface la condición de regularidad y por el 3^{er} caso del teorema maestro concluimos que

$$T(n) = \Theta(f(n))$$

es decir que

$$T(n) = \Theta(4^n)$$

■

- $T(n) = 5T\left(\frac{n}{5}\right) + \frac{n}{\log n}$

Tenemos la función $f(n) = \frac{n}{\log n}$, vimos que no podemos aplicar ningún caso entonces aplicamos sustitución:

Sea $n = 2^m$ así:

$$T(2^m) = 5T\left(\frac{2^m}{5}\right) + \frac{2^m}{\log_2 2^m}$$

$$T(2^m) = 5T\left(\frac{2^m}{5}\right) + \frac{2^m}{m}$$

No encontramos un patrón no obstante tenemos que cuando $f(n) = \Theta(n^{\log_b a} \log^k n)$ para $k = -1$, entonces $T(n) = \Theta(n^{\log_b a} \log \log n)$

por lo que entonces como $f(n) = \Theta\left(n^{\log_5 5} \log^{-1} n\right) = \Theta\left(\frac{n}{\log n}\right)$ entonces tenemos que

$$T(n) = \Theta(n^{\log_5 5} \log \log n) = \Theta(n \log \log n)$$

Y su demostración es la siguiente:

Veamos qué pasa al sustituir:

$$T(n) = 5T\left(\frac{n}{5}\right) + \frac{n}{\log n}$$

$$= 5\left(5T\left(\frac{n}{25}\right) + \frac{\frac{n}{5}}{\log\left(\frac{n}{5}\right)}\right) + \frac{n}{\log n}$$

$$= 25T\left(\frac{n}{25}\right) + \frac{n}{\log\left(\frac{n}{5}\right)} + \frac{n}{\log n}$$

$$\vdots$$

$$* \quad = 5^i T\left(\frac{n}{5^i}\right) + \sum_{j=0}^{i-1} \frac{n}{\log\left(\frac{n}{5^j}\right)}$$

$$\vdots$$

Sin embargo para ver el caso base $T(0)$ igualamos

$$\frac{n}{5^i} = 0 \implies n = 5^i \implies i = \log_5 n$$

Y si sustituimos la i en $*$

$$T(n) = 5^{\log_5 n}\, T\left(\frac{n}{5^{\log_5 n}}\right) + \sum_{j=0}^{\log_5 n - 1} \frac{n}{\log\left(\frac{n}{5^j}\right)}$$

$$= n\,\Theta(1) + \sum_{j=0}^{\log_5 n - 1} \frac{n}{\log\left(\frac{n}{5^j}\right)}$$

$$= \Theta(n) + n \sum_{j=0}^{\log_5 n - 1} \frac{1}{\log_5 n - j \log 5}$$

por propiedades de los logs

$$= \Theta(n) + \frac{n}{\log 5} \sum_{j=0}^{\log_5 n - 1} \frac{1}{\log n - j}$$

$$= \Theta(n) + n \log_5 2 \sum_{k=1}^{\log_5 n} \frac{1}{k}$$

Notemos que tenemos la serie armónica, por lo que entonces concluimos que

$$T(n) = \Theta(n) + n \ln(\log_5 n) = \Theta(n \log\log n)$$

2. (2.5 pts) Se toma un entero positivo a. Dado un entero n, queremos saber si entre los números $a, 2a, 3a, \ldots, na$, hay dos que tengan la misma cantidad de dígitos iguales a 9.

Describe un algoritmo que pueda hacer esto en tiempo $O(n \log n)$. No es necesario que lo implementes ni en código, ni en pseudocódigo. Sin embargo, sí debes discutir su correctitud y explicar por qué corre en el tiempo indicado (ver también Problema 5).

$$\left.\begin{array}{l} 195792 \\ 399772171 \end{array}\right\} \text{Solución Válida}$$

Algoritmo:

1. Dados a y n como entradas, creamos una lista L de tamaño n (en la que guardaremos el número de dígitos 9 de cada número)

2. Definimos un contador i en 1

3. Mientras que nuestro contador sea menor o igual a n, obtenemos el número de dígitos que sean 9 para el número $x = i \cdot a$ y lo guardamos en la lista L

4. Una vez llenada la lista L, la ordenamos.

5. Verificamos si hay dos números iguales consecutivos en L, si hay devolvemos True, si no devolvemos False.

Correctud:

Notemos que es correcto debido a que para cada número obtenemos la cantidad de dígitos que son 9 y almacenamos la cantidad en una lista L, posteriormente se ordena y verificaremos si hay dos números iguales consecutivos, esto nos dice que hubo dos cantidades de 9's para dos números x, y iguales, de ser así el algoritmo devuelve True, sino False.

Complejidad:

1. $O(n)$
2. $O(1)$
3. $O(n)$ Suponiendo que la obtención de los dígitos es constante
4. $O(n \log n)$
5. $O(n)$

Complejidad Total: $O(n \log n)$

3. (2.5 pts) Se tiene un reloj con las 12 posiciones usuales $(1, 2, \ldots, 12)$. Se va a colocar un número del 1 al 24 en cada una de las posiciones, con las siguientes reglas:

- Todos los números que se pusieron son distintos.
- Para cualesquiera dos posiciones que queden una junto a otra, el producto de ambos números es menor que 144.

Se quiere encontrar aquel acomodo (o acomodos) que cumpla ambas condiciones anteriores y en donde la suma de los números que se escribieron en las posiciones sea lo más grande posible.

- Una forma básica de plantear un espacio de estados para este problema es considerar todos los vectores de 12 entradas con números distintos de 1 a 18. ¿Cuántos vectores de estos existen?

vector $\rightarrow$ $(v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12})$

posibilidades $\rightarrow$ 18 17 16 15 14 13 12 11 10 9 8 7

Existen $\frac{18!}{6!}$ vectores en el espacio de estados.

- Imagina que pasar por una posibilidad de las anteriores, verificar si cumple la condición del producto, y verificar si la suma mejora, toma aproximadamente 0.000001 segundos. Aproximadamente, ¿cuánto tiempo tomaría pasar por todos los casos?

Tenemos que en total hay $\frac{18!}{6!}$ casos y si hacer las operaciones a un caso tarda $\frac{1}{1000000}$ s entonces hacer las operaciones para cada caso tardará :

$$\frac{18!}{6!} \cdot \frac{1}{1000000} = \frac{18!}{720\,000\,000} = 8892185.702 \text{ segundos}$$

Es decir aproximadamente 2470 horas!

- Explica cómo usarías una estrategia de simetría para reducir el tiempo anterior en por lo menos un factor de 12. Escribe pseudocódigo para un algoritmo de este estilo.

Para ahorrarnos un factor de 12, dado que si una respuesta será válida entonces sus rotaciones igual serán válidas y notemos que podemos hacer 12 rotaciones. Entonces por cada vector válido habrá otros 11 válidos rotados. Así disminuimos el espacio de estados.