Universidad Simón Bolívar
CI-5651
Enero-Marzo 2024

Tarea 3
Divide y Vencerás

Estudiante: Pedro Ignacio Itzcoatl Rodríguez Reveron
Carnet: 15-11264

1.

a. $T(n) = 3T(\frac{n}{4}) + \frac{3(n^2-1)}{2}$

Sea a=3, b=4 y $g(n) = \frac{3(n^2-1)}{2}$, procedemos a trabajar sobre $g(n)$ para aplicar el teorema maestro.

$O(\frac{3(n^2-1)}{2})$
= Regla de la multiplicación
$O(n^2 - 1)$
= Regla de la suma
$O(n^2)$

Por ende $g(n) \in O(n^2)$ y $k = 2$, por lo que tenemos que se cumple que $a < b^k$, ya que al sustituir obtenemos $3 < 4^2$. Así concluimos que $T(n) \in \theta(n^2)$.

b. $T(n) = 7T(\frac{n}{7}) + 2n - 3$

Sea a=7, b=7 y $g(n) = 2n - 3$, procedemos a trabajar sobre $g(n)$ para aplicar el teorema maestro.

$O(2n - 3)$
= Regla de la suma
$O(2n)$
= Regla de la multiplicación
$O(n)$

Por ende $g(n) \in O(n)$ y $k = 1$, por lo que tenemos que se cumple que $a = b^k$, ya que al sustituir obtenemos $7 = 7$. Así concluimos que $T(n) \in \theta(n\, log\, n)$.

c. $T(n) = 5T(\frac{n}{2}) + 2n$

Sea a=5, b=2 y $g(n) = 2n$, procedemos a trabajar sobre $g(n)$ para aplicar el teorema maestro.

$O(2n)$

= Regla de la multiplicación

$O(n)$

Por ende $g(n) \in O(n)$ y $k = 1$, por lo que tenemos que se cumple que $a > b^k$, ya que al sustituir obtenemos $5 = 2$. Así concluimos que $T(n) \in \theta(n^{log\, 5})$.

d. $T(n) = \frac{\sum_{i=1}^{n}(T(\frac{n}{2})+i)}{n}$

Primero debemos transformar la ecuación antes de aplicar el teorema maestro

$\frac{\sum_{i=1}^{n}(T(\frac{n}{2})+i)}{n}$

= Separación de sumatoria

$\frac{\sum_{i=1}^{n}T(\frac{n}{2})+\sum_{i=1}^{n}i}{n}$

= Fórmula cerrada sumatoria

$\frac{nT(\frac{n}{2})+\frac{n(n+1)}{2}}{n}$

= Álgebra

$T(\frac{n}{2}) + \frac{n+1}{2}$

Ahora obtenemos que a=1, b=2 y $g(n) = \frac{1}{2}(n + 1)$, y procedemos a trabajar sobre $g(n)$.

$O(\frac{n+1}{2})$

= Regla de la multiplicación

$O(n + 1)$

= Regla de la suma

$O(n)$

Por ende $g(n) \in O(n)$ y $k = 1$, por lo que tenemos que se cumple que $a < b^k$, ya que al sustituir obtenemos $1 < 2$. Así concluimos que $T(n) \in \theta(n)$.

2.

Si utilizamos la generalización de recurrencias lineales vista en clases, podemos construir las siguientes matrices:

$$R = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad V = \begin{bmatrix} P(2) \\ P(1) \\ P(0) \end{bmatrix}$$

Tal que definimos la siguiente fórmula o norma que nos permitirá calcular un número de Perrin mediante exponenciaciones y multiplicación de matrices:

$$R^{n-2} \times V = \begin{bmatrix} P(n) \\ P(n-1) \\ P(n-2) \end{bmatrix}$$

Ya con esta fórmula, se propone el siguiente algoritmo:

```
funcion perrin(n: entero) -> entero:
    si (n == 0):
        retornar 3
    
    si (n == 1):
        retornar 0
    
    si (n==2):
        retornar 2

    R = matriz de enteros [0..3][0..3] {inicializada como R}
    V = vector de enteros [0..3]       {inicializado como V}
    retornar (elevar_matriz(R, n-2) x V)[0] // Siendo 0 el primer elemento del vector resultante, P(n)
```

Donde la función *elevelar_matriz* implementa la potencia de una matriz utilizando el algoritmo de exponenciación binaria . Una implementación recursiva de esta función es la siguiente:

```
funcion elevar_matriz(mat: matriz de enteros[0..n][0..n], pow: entero):
    si (pow == 0):
        retornar (matriz identidad [0..n][0..n])

    p = elevar_matriz(mat, pow / 2)
    si (pow mod 2 == 0):
        retornar p * p
    
    retornar a * p * p
```

El algoritmo entonces consiste en:

- Si n es un caso base, devolver el valor de este caso.

- Si no, calcular la n-2 potencia de la matriz R, multiplicarla por el vector V, y retornar el primer elemento que es *P(n)*.

Respecto a la complejidad asintótica de este algoritmo, realizamos el siguiente análisis:

- Calcular los casos bases es tiempo constante $\Theta(1)$.

- La matriz R y V pueden ser constantes, por lo que su creación es $\Theta(1)$.

- Respecto a la función *elevar_matriz*, se usa una recursión con la técnica de divide y vencerás con una división del subproblema a la mitad, por lo que aplicaremos el teorema maestro para calcular su complejidad. Quedamos así con $T(n) = T(\frac{n}{2}) + g(n)$ con la división *n/2* calculada con piso, donde debido a la suposición de que todas las operaciones aritméticas son $\Theta(1)$ entonces $g(n) \in \Theta(1)$ y podemos así $a = 1,\ b = 2,\ k = 0$ por lo que $a = b^k$ ya que al reemplazar los valores obtenemos $1 = 1$, concluyendo así que $elevar\ matriz \in \Theta(log\ n)$.

- Se asume que la multiplicación es orden constante, por lo que en el algoritmo $R^{n-2} \times V \in \Theta(1)$.

Por ende, se concluye que el algoritmo propuesto pertenece al orden $\Theta(log\ n)$.