Actividades sesión 12 de octubre
Análisis de Algoritmos IV
Maestría en Ciencias de la Computación
Otoño 2020
BUAP

Equipo 3
Erick Barrios González
Oscar Eduardo González Ramos
Oswaldo Jair García Franco

15 de octubre de 2020

1. Ejercicios

Resuelva las siguientes recurrencias exactamente,perimero mediante cambio de variable, luego verifique sus respuestas empleando el resultado del ejemplo 4.

1

$$T(n) = \begin{cases} 1 & \text{si n = 1} \\ 2T(n/2) + 1 & \text{en otro caso} \end{cases}$$

Reemplazamos n por 2^i:
$t_i = T(2^i)$

$n/2$ se convierte en :
$\frac{2^i}{2} = 2^{i-1}$

Entonces:
$$t_i = T(2^i) = 2T(2^{i-1}) + 1$$
$$= 2t_{i-1} + 1$$

Se reescribe como:
$t_i - 2t_{i-1} = 1$

Que es la forma de una recurrencia no homogénea
Su polinomio característico es:
$(x-2)(x-1)$

Así todas las soluciones para t_i son de la forma:
$t_i = c_1(2^i) + c_2$

Ahora regresamos a la ecuación original
usamos el hecho de que $T(2^i) = t_i$, Así $T(n) = tlgn$
$T(n) = c_1 2^{lgn} + c2 = c_1 n + c_2$ (1)

Lo cual es suficiente para concluir que :
$T(n) \in O(n|n$ es potencia de 2)
Como queremos obtener el orden exacto debemos probar que c_1 es estrictamente positivo.

Sustituyendo la solución proporcionada por la ecuación (1) en la recurencia original.
$$1 = T(n) - 2T(n/2)$$
$$= (c_1 n + c2) - 2(c_1 \frac{n}{2} + c_2)$$
$$c_2 = -1$$
No obtuvimos el valor de c_1 pero estamos en condiciones de afirmar que debe ser estrictamente positiva, de lo contrario la ecuación (1) indicaría

falsamente que $T(n)$ es negativa.
Entonces queda establecido que:
$T(n) \in \Theta(n|n$ es potencia de 2)

Comprobación con teorema Maestro: $l = 2$, $b = 2$,$k = 0$
$l > 2^0 = 1$
Entonces $T(n) \in \Theta(n^{\log_2(2)})$
$T(n) \in \Theta(n|n$ es potencia de 2)

2

$$T(n) = \begin{cases} 1 & \text{si n = 1} \\ 4T(n/2) + n & \text{en otro caso} \end{cases}$$

Reemplazamos n por 2^i:
$t_i = T(2^i)$

$n/2$ se convierte en :
$\frac{2^i}{2} = 2^{i-1}$

Entonces:
$$t_i = T(2^i) = 4T(2^{i-1}) + 2^i$$
$$= 4t_{i-1} + 2^i$$

Se reescribe como:
$t_i - 4t_{i-1} = 2^i$
Que es la forma de una recurrencia no homogénea

Su polinomio característico es:
$(x-4)(x-2)$

Así todas las soluciones para t_i son de la forma:
$t_i = c_1(4^i) + c_2(2^i)$

Ahora regresamos a la ecuación original
usamos el hecho de que $T(2^i) = t_i$, Así $T(n) = tlgn$
$T(n) = c_1 4^{lgn} + c_2 2^{lgn} = c_1 n^2 + c_2 n$ (1)

Lo cual es suficiente para concluir que :
$T(n) \in O(n^2 | n \text{ es potencia de } 2)$

Como queremos obtener el orden exacto debemos probar que c_1 es estrictamente positivo.
Sustituyendo la solución proporcionada por la ecuación (1) en la recurencia original.
$$n = T(n) - 4T(n/2)$$
$$= (c_1 n^{lg4} + c_2 n) - 4(c_1(\tfrac{n}{2})^{lg4} + c_2(\tfrac{n}{2}))$$
$$c_2 = -1$$
No obtuvimos el valor de c_1 pero estamos en condiciones de afirmar que debe ser estrictamente positiva, de lo contrario la ecuación (1) indicaría falsamente que $T(n)$ es negativa.
Entonces queda establecido que:
$T(n) \in \Theta(n^2 | n \text{ es potencia de } 2)$

Comprobación con teorema Maestro: $l = 4$, $b = 2$,$k = 0$

$l > 2^0 = 1$

Entonces $T(n) \in \Theta(n^{\log_2(4)})$

$T(n) \in \Theta(n^2 | n$ es potencia de 2)

3

$$T(n) = \begin{cases} 1 & \text{si n = 1} \\ 2T(n/2) + lgn & \text{en otro caso} \end{cases}$$

Reemplazamos n por 2^i:

$t_i = T(2^i)$

$n/2$ se convierte en :

$\frac{2^i}{2} = 2^{i-1}$

Entonces:

$$\begin{aligned} t_i = T(2^i) &= 2T(2^{i-1}) + lg(2^i) \\ &= 2t_{i-1} + ilg2 \\ &= 2t_{i-1} + i \end{aligned}$$

Se reescribe como:

$t_i - 2t_{i-1} = i$

Su polinomio característico es:

$(x-2)(x-1)$

Así todas las soluciones para t_i son de la forma:

$t_i = c_1 2^i + c_2 + c_3 i$

Ahora regresamos a la ecuación original

$T(n) = c_1 n + c_2$

Como queremos obtener el orden exacto debemos probar que c_1 es estrictamente positivo.

Sustituyendo la solución proporcionada por la ecuación (1) en la recurencia original.

$$\begin{aligned} lgn &= T(n) - 2T(n/2) \\ &= (c_1 n + c_2) - 2(c_1(\tfrac{n}{2}) + c_2) \\ c_2 &= -logn \end{aligned}$$

No obtuvimos el valor de c_1 pero estamos en condiciones de afirmar que debe ser estrictamente positiva Por lo tanto :

$T(n) \in O(n | n \text{ es potencia de } 2)$

Comprobación con teorema Maestro: $l = 2$, $b = 2$,$k = 0$

$l > b^k$

Porque: $2 > 2^0$

Por lo tanto:

$T(n) \in \Theta(n^{\log_2(2)})$

$T(n) \in \Theta(n | n \text{ es potencia de } 2)$

4

$$T(n) = \begin{cases} 1 & \text{si n = 1} \\ 5T(n/2) + (nlgn)^2 & \text{en otro caso} \end{cases}$$

Reemplazamos n por 2^i:
$t_i = T(2^i)$

$n/2$ se convierte en :
$\frac{2^i}{2} = 2^{i-1}$

Entonces:
$t_i = T(2^i) = 5T(2^{i-1}) + (2^i lg(2^i))^2$
$= 5T(2^{i-1}) + (2^i i)^2$

Se reescribe como:
$t_i - 5t_{i-1} = i^2 4^i$

Su polinomio característico es:
$(x-5)(x-4)$

Así todas las soluciones para t_i son de la forma:
$t_i = c_1 5^i + c_2 4^i$

Ahora regresamos a la ecuación original
$T(n) = c_1 n^{lg5} + c_2 n^{lg4}$

Como queremos obtener el orden exacto debemos probar que c_1 es estrictamente positivo.
Sustituyendo la solución proporcionada por la ecuación (1) en la recurencia original.
$(nlgn)^2 = T(n) - 5T(n/2)$
$= (c_1 n^{lg5} + c_2 n^{lg4}) - 2(c_1(\frac{n}{2})^{lg5} + c_2(\frac{n}{2})^{lg4})$
$c_2 n^{lg4} = -(nlogn)^2/4$
Como c_2 es negativa estamos en condiciones de afirmar que c_1 debe ser estrictamente positiva Por lo tanto :
$T(n) \in O(n^{log5}|n$ es potencia de 2)

Comprobación con teorema Maestro: $l = 5$, $b = 2$,$k = 0$
$l > b^k$
Porque: $5 > 2^0$
Por lo tanto:
$T(n) \in \Theta(n^{\log_2(5)})$
$T(n) \in \Theta(n^{\log_2(5)}|n$ es potencia de 2)

5

$$T(n) = \begin{cases} 1 & \text{si n = 1} \\ 2T(n/3) + 4 & \text{en otro caso} \end{cases}$$

Reemplazamos n por 3^i:

$$t_i = T(3^i)$$
$$t_i = T(3^i) = 2T(3^{i-1}) + 4$$

Se reescribe como:

$$t_i - 2t_{i-1} = 4$$

Su polinomio característico es:

$$(x-2)(x-4)$$

Así todas las soluciones para t_i son de la forma:

$$t_i = c_1 2^i + c_2 4^i$$

Ahora regresamos a la ecuación original
usando el hecho de que $T(3^i) = t_i$, así $T(n) = t \log n$

$$T(n) = C_1 2^{lgn} + C_2 4^{lgn}$$
$$T(n) = C_1 n^{lg2} + C_2 n^{lg4} \text{ (1)}$$

Como queremos obtener el orden exacto debemos probar que c_2 es estrictamente positivo.
Sustituyendo la solución proporcionada por la ecuación (1) en la recurencia original.

$$4 = T(n) - 2T(n/3)$$
$$4 = (c_1 n^{lg2} + c_2 n^{lg4}) - 2(c_1(\tfrac{n}{3})^{lg2} + c_2(\tfrac{n}{2})^{lg4})$$
$$c_2 = (\tfrac{n}{2})^{lg4}$$

Por lo tanto se puede concluir que :

$$T(n) \in O(n^{lg4} | n \text{ es potencia de 3})$$

Comprobación con teorema Maestro: $l = 2$, $b = 3$,$k = 0$

$$l > b^k$$

Porque: $2 > 3^0$

Por lo tanto:

$$T(n) \in O(n^{\log_3(2)})$$
$$T(n) \in O(n^{\log_3(2)} | n \text{ es potencia de 3})$$

6

$$T(n) = \begin{cases} 1 & \text{si n} = 1 \\ T(n/3) + 2 & \text{en otro caso} \end{cases}$$

Reemplazamos n por 3^i:
$t_i = T(3^i)$
$t_i = T(3^i) = T(3^{i-1}) + 2$

Se reescribe como:
$t_i - t_{i-1} = 2$

Su polinomio característico es:
$(x-1)(x-2)$

Así todas las soluciones para t_i son de la forma:
$t_i = c_1 1 + c_2 2^i$

Ahora regresamos a la ecuación original
usando el hecho de que $T(3^i) = t_i$, así $T(n) = t \log n$
$T(n) = C_1 1^{lgn} + C_2 2^{lgn} = C_1 + C_2 n^{lg2}$ (1)

Como queremos obtener el orden exacto debemos probar que c_2 es estrictamente positivo.
Con el sistema
$C_1 + C_2 3^{lg3^2}) = 3, n = 3$
$C_1 = 0$
$C_2 = 1$
$4 = (c_1 n^{lg2} + c_2 n^{lg4}) - 2(c_1(\frac{n}{3})^{lg2} + c_2(\frac{n}{2})^{lg4})$
$c_2 = (\frac{n}{2})^{lg4}$
Por lo tanto se puede concluir que :
$T(n) \in O(n^{lg2} | n$ es potencia de 3)

Comprobación con teorema Maestro: $l = 1$, $b = 3$,$k = 0$
$l = b^k$
Porque: $1 = 3^0$
Por lo tanto:
$T(n) \in \Theta(n^0 \log n) = \Theta(\log n)$

7

$$T(n) = \begin{cases} 1 & \text{si n = 2} \\ 2T(\sqrt{n}) + lgn & \text{en otro caso} \end{cases}$$

Reemplazamos n por 2^i:

$t_i = T(2^i)$

$t_i = T(2^i) = 2T(2^{\frac{i}{2}}) + i$

$S(\frac{i}{2}) = T(2^{\frac{1}{2}})$

Siendo la forma que podemos resolver:

Sustituimos i por 2^i

$Si = 2S(2^{i-1}) + i$

$Si - 2Si - 1 = i$

Su polinomio característico es:

$(x-2)(x-1)^2$

$S(2^i) = Si$

Con soluciones de la forma:

$S(n) = C_1 2^i + C_2 + C_3 i$

$T(n) = \log n \log\log n$

Ahora regresamos a la ecuación original

$T(n) = C_1 2^{\log n \log\log n} + C_2 + C_3 \log n \log\log n$

$T(n) = C_1 n \log\log n + C_2 + C_3 \log n \log(\log n)$

Por lo tanto se puede concluir que :

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

Comprobación con teorema Maestro:

$T(n) = 2T(\sqrt{n}) + \log n$

$m = \log n$ producciones

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

Luego $S(m) = T(2^m)$ produce la recurrencia

$S(m) = 2S(\frac{m}{2}) + m$

Entonces $T(n) = T(2^m) = S(m) = O(m \log m) = O(\log n(\log\log n))$

8 Teorema Maestro

El teorema maestro proporciona una manera de resolver recurrencias de la forma:

$$T(n) = aT(\frac{n}{b}) + f(n)$$

Donde $a \geq 1$ y $b > 1$ son constantes y $f(n)$ es una función asintóticamente positiva.
Está recurrencia describe el tiempo de un algoritmo que:

- Divide un problema de tamaño n en a subproblemas, cada uno de tamaño $\frac{n}{b}$, donde a y b son constantes positivas.
- Los a subproblemas se resuelven recursivamente, cada uno en un tiempo $T(\frac{n}{b})$.
- El costo de dividir el problema y combinar los resultados de los subproblemas está descrito por la función $f(n)$.

Entonces $T(n)$ puede estar acotada asintóticamente de la forma siguiente:
3 casos:

1. Si $f(n) = O(n^{\log_b a - \epsilon})$ para alguna constante $\epsilon > 0$ entonces

$$T(n) = \Theta(n^{\log_b a})$$

2. Si $f(n) = \Theta(n^{\log_b a})$, entonces:

$$T(n) = \Theta(n^{\log_b a} \log n)$$

3. Si $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguna constante $\epsilon > 0$, y si $af(\frac{n}{b}) \leq cf(n)$ para alguna constante $c < 1$ y una n suficientemente grande, entonces:

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

En los tres casos se compara la función $f(n)$ con la función $n^{\log_b a}$
Intuitivamente, la solución a la recurrencia estará determinada por la mayor de las funciones.

- En 1. La función $n^{\log_b a}$ crece más rápido polinomialmente que $f(n)$ por un factor n^{ϵ}. Además, $f(n)$ satisface la condición de regularidad que $f(n)$ por un factor n^{ϵ}. La solución esta dada entonces por $n^{\log_b a}$
- En 3. La función $f(n)$ crece polinomialmente más rápido que $n^{\log_b a}$ por un factor n^{ϵ}. Además $f(n)$ satisface la condición de regularidad que dice $af(\frac{n}{b}) \leq cf(n)$ para alguna constante $c > 1$.
- Los tres casos no cubren todas las posibilidades, el método no puede aplicarse si las funciones no crecen polinomialmente más rápido o si la condición de regularidad no se cumple.

Referencias

[1] Lavalle Martínez, José de Jesús; La presentación sobre la tercera parte de Análisis de algoritmos, Análisis y Diseño de Algoritmos. Buap, Otoño 2020.

[2] Jiménez Salazer, Héctor y Lavalle Martínez, José de Jesús; Análisis y Diseño de Algoritmos. Traducción de partes del libro Fundamentals of Algorithmics de Brassard and Bratley FCC - BUAP, 2020.