Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Análise de Algoritmos
Lista de Exercícios – Relações de Recorrência
Prof. Daniel Saad Nogueira Nunes

Aluno: ____________________
Matrícula: ____________________

Exercício 1

Demonstre as questões abaixo por indução matemática.

(a) Mostre que a série abaixo possui esta desigualdade

$$\sum_{i=1}^{n} \frac{1}{2^i} < 1$$

Solução:
O caso base é facilmente verificável:

$$\sum_{i=1}^{1} \frac{1}{2^i} = \frac{1}{2} < 1$$

Hipótese de indução: $\forall k$, com $1 \leq k \leq n$

$$\sum_{i=1}^{k} \frac{1}{2^i} < 1$$

Vamos mostrar agora que:

$$\sum_{i=1}^{n+1} \frac{1}{2^i} < 1$$

$$\begin{aligned}
\sum_{i=1}^{n+1} \frac{1}{2^i} =& \frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^n} + \frac{1}{2^{n+1}} \\
=& \frac{1}{2} + \underbrace{\frac{1}{2} \cdot \underbrace{\left(\frac{1}{2} + \ldots + \frac{1}{2^{n-1}} + \frac{1}{2^n} \right)}_{<1}}_{<\frac{1}{2}} \\
\therefore \sum_{i=1}^{n+1} \frac{1}{2^i} < 1
\end{aligned}$$

(b) Demonstre que $x^n - y^n$ é divisível por $x - y$, para todos os números naturais x, y com $x \neq y$ e $n \in \mathbb{N}$.

Solução:
O caso base é facilmente verificável. Quando $n = 1$:

$$(x - y) \mid (x - y)$$

Hipótese de indução:

$$(x - y) \mid (x^n - y^n)$$

Vamos mostrar agora que:

$$(x - y) \mid (x^{n+1} - y^{n+1})$$

Primeiramente:

$$\begin{aligned}
x^{n+1} - y^{n+1} =& x^{n+1} - x^n y + x^n y - y^{n+1} \\
=& \underbrace{x^n(x - y)}_{\text{H.I: divisível por (x-y)}} + \underbrace{y(x^n - y^n)}_{\text{H.I: divisível por (x-y)}} \\
\therefore (x - y) \mid (x^{n+1} + y^{n+1})
\end{aligned}$$

(c) O princípio das casas dos pombos afirma que, se temos $n+1$ bolas $n \geq 1$ e n caixas, e colocarmos todas as bolas nas caixas, pelo menos uma caixa terá mais que uma bola. Demonstre esse princípio.

Solução:
O caso base é facilmente verificável, quando temos 2 bolas 1 caixa, a primeira caixa contém 2 bolas.

Hipótese de indução: dada $n+1$ bolas e n caixas, pelo menos uma caixa terá 2 bolas.

Vamos mostrar agora que se tivermos $n + 2$ bolas e $n + 1$ caixas, pelo menos uma caixa ficará com 2 bolas.

- Caso 1: a caixa $n+1$ tem pelo menos duas bolas. Estamos prontos.
- Caso 2: a caixa $n+1$ tem exatamente uma bola. Logo, sobram $n+1$ bolas para as n primeiras caixas, e pela hipótese de indução, pelo menos uma das n primeiras caixas terá mais que 1 bola.
- Caso 3; a caixa $n+1$ não possui nenhuma bola. Logo, sobram $n+2 > n+1$ bolas para as n primeiras caixas, e pela hipótese de indução, pelo menos uma das n primeiras caixas terá mais que 1 bola.

(d) Mostre que o termo geral de uma progressão aritmética é dado por:

$$a_n = a_0 + rn$$

Solução:
Caso base:

$$a_0 = a_0$$

Hipótese de indução:

$$a_{n-1} = a_0 + r(n-1)$$

Vamos mostrar agora que

$$a_n = a_0 + rn$$

$$\begin{aligned} a_n =& a_{n-1} + r \\ =& \underbrace{a_0 + r(n-1)}_{\text{H.I}} + r \\ =& a_0 + rn \end{aligned}$$

(e) Mostre que a soma de uma progressão aritmética é dada por:

$$S_n = \frac{n(a_0 + a_{n-1})}{2}$$

Solução:
Caso base:

$$S_1 = \frac{1 \cdot (a_0 + a_0)}{2} = a_0$$

Hipótese de indução:

$$S_{n-1} = \frac{(n-1) \cdot (a_0 + a_{n-2})}{2}$$

Vamos mostrar agora que:

$$S_n = \frac{n(a_0 + a_{n-1})}{2}$$

Primeiramente:

$$S_n = S_{n-2} + a_{n-1}$$

$$\begin{aligned}
S_n =& S_{n-2} + a_{n-1} \\
=& \frac{(n-1)}{2}(a_0 + a_{n-2}) + a_{n-1} \\
=& \frac{(n-1)(a_0 + a_{n-2}) + 2a_{n-1}}{2} \\
=& \frac{na_0 - a_0 + na_{n-2} - a_{n-2} + 2a_{n-1}}{2} \\
=& \frac{na_0 - a_0 + n(a_0 + r(n-2)) - a_0 - r(n-2) + 2(a_0 + r(n-1))}{2} \\
=& \frac{na_0 - a_0 + na_0 + nrn - 2rn - a_0 - rn + 2r + 2a_0 + 2rn - 2r}{2} \\
=& \frac{2na_0 + nrn - rn}{2} \\
=& \frac{n(2a_0 + r(n-1)}{2} \\
=& \frac{n(a_0 + a_0 + r(n-1)}{2} \\
=& \frac{n(a_0 + a_{n-1})}{2}
\end{aligned}$$

(f) Mostre que o termo geral de uma progressão geométrica é dada por:

$$a_n = a_0 r^n$$

Solução:

Caso base:

$$a_0 = a_0 r^0 = a_0$$

Hipótese de Indução:

$$a_{n-1} = a_0 r^{n-1}$$

Vamos mostrar agora que:

$$a_n = a_0 + r^n$$

Primeiramente:

$$a_n = a_{n-1} r$$

$$
\begin{aligned}
a_n =& a_{n-1} r \\
=& a_0 r^{n-1} r \\
=& a_0 r^n
\end{aligned}
$$

(g) Mostre que a soma de uma progressão aritmética, com $r \neq 1$ é dada por:

$$S_n = \frac{a_0(r^n - 1)}{r - 1} = \frac{a_0(1 - r^n)}{1 - r}$$

Solução:
Caso base:

$$S_1 = \frac{a_0(r - 1)}{(r - 1)} = a_0$$

Hipótese de Indução:

$$S_{n-1} = \frac{a_0(r^{n-1} - 1)}{r - 1}$$

Vamos mostrar que:

$$S_n = \frac{a_0(r^n - 1)}{r - 1}$$

Primeiramente:

$$S_n = S_{n-1} + a_{n-1}$$

$$
\begin{aligned}
S_n =& S_{n-1} + a_{n-1} \\
=& \frac{a_0(r^{n-1} - 1)}{r - 1} + a_{n-1} \\
=& \frac{a_0 r^{n-1} - a_0 + a_{n-1}(r - 1)}{r - 1} \\
=& \frac{a_0 r^{n-1} - a_0 + a_{n-1} r - a_{n-1}}{r - 1} \\
=& \frac{a_0 r^{n-1} - a_0 + a_0 r^{n-1} r - a_0 r^{n-1}}{r - 1} \\
=& \frac{-a_0 + a_0 r^n}{r - 1} \\
=& \frac{a_0(r^n - 1)}{r - 1}
\end{aligned}
$$

Exercício 2

Prove que $T(n) \in \Theta(n^2)$ de modo que:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(n-1) + \Theta(n), & n > 1 \end{cases}$$

Solução:

Primeiramente provaremos que $T(n) \in O(n^2)$, isto é:

$$T(n) \leq cn^2$$

para algum $c > 0$ e a partir de um n_0 suficientemente grande.

Pelo método da substituição:

$$\begin{aligned} T(n) \leq & c(n-1)^2 + an \\ =& cn^2 - 2cn + an \\ =& cn^2 - 2cn + an \end{aligned}$$

Precisamos avaliar as condições de existência de c para que

$$cn^2 - 2cn + an \leq cn^2$$

$$\begin{aligned} & cn^2 - 2cn + an \leq cn^2 \\ & -2cn + an \leq 0 \\ & -2cn \leq -an \\ & 2cn \geq an \\ & c \geq \frac{a}{2} \end{aligned}$$

Portanto $T(n) \in O(n^2)$

Provaremos agora que $T(n) \in \Omega(n^2)$, isto é:

$$T(n) \geq cn^2$$

para algum $c > 0$ e a partir de um n_0 suficientemente grande.

Pelo método da substituição:

$$\begin{aligned} T(n) \geq & c(n-1)^2 + an \\ =& cn^2 - 2cn + an \\ =& cn^2 - 2cn + an \end{aligned}$$

Precisamos avaliar as condições de existência de c para que

$$cn^2 - 2cn + an \geq cn^2$$

$$\begin{aligned}
&cn^2 - 2cn + an \geq cn^2 \\
&-2cn + an \geq 0 \\
&-2cn \geq -an \\
&2cn \leq an \\
&c \leq \frac{a}{2}
\end{aligned}$$

Portanto $T(n) \in \omega(n^2)$.

Como $T(n) \in O(n^2)$ e $T(n) \in \Omega(n^2)$, então $T(n) \in \Theta(n^2)$

Exercício 3

Prove pelo método da substituição que $T(n) \in \Theta(\log n)$ de modo que:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(\lfloor n/2 \rfloor) + \Theta(1), & n > 1 \end{cases}$$

Solução: Primeiramente mostraremos que $T(n) \in O(\log n)$, isto é:

$$T(n) \leq c \log n$$

para algum $c > 0$ e a partir de um n_0 suficientemente grande.

Pelo método da substituição:

$$\begin{aligned}
T(n) \leq& c\left(\log\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right) + a \\
\leq& c\left(\log\left(\frac{n}{2}\right)\right) + a \\
=& c(\log(n) - \log(2)) + a \\
=& c\log(n) - c\log(2) + a
\end{aligned}$$

Precismos mostrar agora a condição de existência de c para que:

$$c\log(n) - c\log(2) + a \leq c \log n$$

$$
\begin{aligned}
&c\log(n) - c\log(2) + a \leq c\log n \\
&- c\log(2) \leq -a \\
&c\log(2) \geq a \\
&c \geq \frac{a}{\log(2)}
\end{aligned}
$$

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

Agora mostraremos que $T(n) \in \Omega(\log n)$, isto é:

$$T(n) \geq c\log n$$

para algum $c > 0$ e a partir de um n_0 suficientemente grande.

Pelo método da substituição:

$$
\begin{aligned}
T(n) \geq & c\left(\log\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right) + a \\
\geq & c\left(\log\left(\frac{n}{2} - 1\right)\right) + a \\
\geq & c\left(\log\left(\frac{n}{2} - \frac{n}{4}\right)\right) + a \\
\geq & c\left(\log\left(\frac{n}{4}\right)\right) + a \\
= & c(\log(n) - \log(4)) + a \\
= & c\log(n) - c\log(4) + a
\end{aligned}
$$

Precismos mostrar agora a condição de existência de c para que:

$$c\log(n) - c\log(4) + a \geq c\log n$$

$$
\begin{aligned}
&c\log(n) - c\log(4) + a \geq c\log n \\
&- c\log(4) \geq -a \\
&c\log(4) \leq a \\
&c \leq \frac{a}{\log(4)}
\end{aligned}
$$

Portanto, $T(n) \in \Omega(\log n)$.

Como $T(n) \in O(\log n)$ e $T(n) \in \Omega(\log n)$ então, $T(n) \in \Theta(\log n)$.

Exercício 4

Demonstre que a busca binária possui tempo de pior caso $\Theta(\log n)$.

Solução:

De acordo com o algoritmo de busca binária, sempre sobram ao máximo $\lfloor \frac{n}{2} \rfloor$ elementos na próxima chamada recursiva. Portanto o custo de tal algoritmo pode ser modelado via a relação de recorrência:

$$T(n) = \begin{cases} \Theta(1), \quad n = 1 \\ T(\lfloor n/2 \rfloor) + \Theta(1), \quad n > 1 \end{cases}$$

Esta relação $T(n)$, conforme exercício anterior, possui cota $T(n) \in \Theta(\log n)$.

Algorithm 1: BSEARCH(V, k, l, r)

Input: V,k,l,r
Output: $j, \quad V[j] = k$

```
if l > r then
    return key-not-found;
else
    mid = (l+r)/2;
    if k = V[mid] then
        return mid;
    else if k < V[mid] then
        return BSEARCH(V, k, l, mid - 1);
    else
        return BSEARCH(V, k, mid + 1, r);
```

Exercício 5

Ache uma cota assintoticamente justa para:

$$T(n) = \begin{cases} \Theta(1), \quad n = 1 \\ 3T(n/4) + \Theta(n^2), \quad n > 1 \end{cases}$$

Solução:

Mostraremos que $T(n) \in O(n^2)$ primeiramente, isto é:

$$T(n) \leq cn^2$$

para algum $c > 0$ e a partir de $n \leq n_0$ suficientemente grande.

Pelo método da substituição:

$$
\begin{aligned}
T(n) \leq & 3c(\frac{n}{4})^2 + an^2 \\
= & \frac{3cn^2}{16} + an^2
\end{aligned}
$$

Precisamos avaliar as condições de existência de c para que:

$$
\frac{3cn^2}{16} + an^2 \leq cn^2
$$

$$
\begin{aligned}
& \frac{3cn^2}{16} + an^2 \leq cn^2 \\
& an^2 \leq \frac{13cn^2}{16} \\
& c \geq \frac{16a}{13}
\end{aligned}
$$

Portanto, $T(n) \in O(n^2)$.

Mostraremos agora que $T(n) \in \Omega(n^2)$, isto é:

$$
T(n) \geq cn^2
$$

para algum $c > 0$ e a partir de um n_0 suficientemente grande.

Pelo método da substituição:

$$
\begin{aligned}
T(n) \geq & 3c(\frac{n}{4})^2 + an^2 \\
= & \frac{3cn^2}{16} + an^2
\end{aligned}
$$

Precisamos avaliar as condições de existência de c para que:

$$
\frac{3cn^2}{16} + an^2 \geq cn^2
$$

$$
\begin{aligned}
& \frac{3cn^2}{16} + an^2 \geq cn^2 \\
& an^2 \geq \frac{13cn^2}{16} \\
& c \leq \frac{16a}{13}
\end{aligned}
$$

Portanto, $T(n) \in \Omega(n^2)$.

Como $T(n) \in O(n^2)$ e $T(n) \in \Omega(n^2)$, temos que $T(n) \in \Theta(n^2)$.

Exercício 6

Prove que $T(n) \in \Theta(n \log n)$ de modo que:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 2T(\lfloor n/2 \rfloor + 17) + \Theta(n), & n > 1 \end{cases}$$

Solução:

Mostraremos que $T(n) \leq c(n-a)n\log(n-a)$, que claramente é $O(n \log n)$.

Pelo método da substituição:

$$\begin{aligned}
T(n) \leq & 2c\left(\left\lfloor \frac{n}{2} \right\rfloor + 17 - a\right) \log\left(\left\lceil \frac{n}{2} \right\rceil + 17 - a\right) + dn \\
\leq & 2c\left(\frac{n}{2} + 17 - a\right) \log\left(\frac{n}{2} + 17 - a\right) + dn \\
= & c(n + 34 - 2a) \log\left(\frac{n + 34 - 2a}{2}\right) + dn \\
= & c(n + 34 - 2a)(\log(n + 34 - 2a) - \log 2) + dn \\
\leq & c(n - a)(\log(n - a) - \log(2)) + dn \qquad \diamond \text{ quando } a > 34
\end{aligned}$$

Precisamos agora verificar a condição de existência de c para que:

$$c(n-a)(\log(n-a) - \log(2)) + dn \leq c(n-a)$$

$$\begin{aligned}
& c(n-a)(\log(n-a) - \log(2)) + dn \leq c(n-a) \\
& -c(n-a)(\log 2) + dn \leq 0 \\
& dn \leq c(n-a)(\log(2)) \\
& c \geq \frac{dn}{(n-a)(\log(2))} \\
& c \geq \frac{dn}{n\log(2) - a\log(2)} \\
& c \geq \frac{dn}{n \log 2 - \frac{an\log(2)}{n}} \\
& c \geq \frac{d}{\log(2) - \frac{a \log 2}{n}} \\
& c \geq \frac{d}{\log(2)}, \qquad \diamond \text{ com } n >> a
\end{aligned}$$

Portanto $T(n) \leq c(n-a)\log(n-a)$ com $a > 34$ e $c \geq \frac{d}{\log 2}$ com n suficientemente grande.

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

Mostraremos agora que $T(n) \leq cn \log n$, isto é, $T(n) \in O(n \log n)$

Pelo método da substituição:

$$
\begin{aligned}
T(n) \geq & 2c\left(\left\lfloor \frac{n}{2} \right\rfloor + 17\right) \log \left(\left\lceil \frac{n}{2} \right\rceil + 17\right) + dn \\
\geq & 2c\left(\frac{n}{2} - 1 + 17\right) \log \left(\frac{n}{2} - 1 + 17\right) + dn \\
= & 2c\left(\frac{n}{2} + 16\right) \log \left(\frac{n}{2} + 16\right) + dn \\
\geq & 2c\left(\frac{n}{2}\right) \log \left(\frac{n}{2}\right) + dn \\
= & cn \log \left(\frac{n}{2}\right) + dn = & cn \log(n) - cn \log(2) + dn
\end{aligned}
$$

Precisamos avaliar a condição de existência de c para que:

$$cn \log(n) - cn \log(2) + dn \geq cn \log(n)$$

$$
\begin{aligned}
& cn \log(n) - cn \log(2) + dn \geq cn \log(n) \\
& -cn \log(2) + dn \geq 0 \\
& dn \geq cn \log(2) \\
& c \leq \frac{dn}{\log(2) n} \\
& c \leq \frac{d}{\log(2)}
\end{aligned}
$$

E portanto $T(n) \geq cn \log(n)$ com n suficientemente grande e $c \leq \frac{d}{\log(2)}$.

Assim $T(n) \in \Omega(n \log n)$.

Como $T(n) \in O(n \log n)$ e $T(n) \in \Omega(n \log n)$ então $T(n) \in \Theta(n \log n)$

Exercício 7

Prove pelo método mestre que a seguinte relação de recorrência

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 4T(n/3) + n, & n > 1 \end{cases}$$

pertence à $\Theta(n^{\log_3 4})$.

Tente mostrar o mesmo pelo método da substituição e veja que o "chute" $T(n) \leq cn^{\log_3 4}$ irá falhar, sendo necessário reforçar a **hipótese** de indução.

Solução:

Pelo método mestre, temos que:

$n \in O(n^{\log_3 4-\epsilon})$, com $\epsilon < \log_3 4 - 1$

Portanto $T(n) \in \Theta(n^{\log_3 4})$.

Caso viéssemos a mostrar pelo método da substituição teríamos:

$$
\begin{aligned}
T(n) \leq & 4c\left(\frac{n}{3}\right)^{\log_3 4} + n \\
= & 4c\frac{n^{\log_3 4}}{3^{\log_3 4}} + n \\
= & 4c\frac{n^{\log_3 4}}{4} + n \\
= & cn^{\log_3 4} + n \nleq cn^{\log_3 4}
\end{aligned}
$$

Precisamos reforçar a hipótese de indução, isto é, temos que provar que: $T(n) \leq c(n^{\log_3 4} - c2n)$, que é claramente $O(n^{\log_3 4})$.

Pelo método da substituição, temos:

$$
\begin{aligned}
T(n) \leq & 4c\left(\frac{n}{3}^{\log_3 4} - \frac{n}{3}\right) + n \\
= & cn^{\log_3 4} - \frac{4n}{3} + n
\end{aligned}
$$

Precisamos agora achar a condição de existência de c para que:

$$
cn^{\log_3 4} - c\frac{4n}{3} + n \leq cn^{\log_3 4} - cn
$$

$$
\begin{aligned}
& cn^{\log_3 4} - c\frac{4n}{3} + n \leq cn^{\log_3 4} - cn \\
& - c\frac{n}{3} + n \leq 0 \\
& c \geq \frac{n}{\frac{n}{3}} \\
& c \geq 3
\end{aligned}
$$

E portanto, com $c \geq 3$ e n suficientemente grande, temos que $T(n)\ inO(n^{\log_3 4})$.

Mostraremos agora que :

$$
T(n) \geq cn^{\log_3 4}
$$

Pelo método da substituição:

$$\begin{aligned}
T(n) \geq& 4c\left(\frac{n}{3}\right)^{\log_3 4} + n \\
=& 4c\frac{n^{\log_3 4}}{3^{\log_3 4}} + n \\
=& 4c\frac{n^{\log_3 4}}{4} + n \\
=& cn^{\log_3 4} + n \geq cn^{\log_3 4}
\end{aligned}$$

Para $c > 0$ e n suficientemente grande. Assim $T(n) \in \Omega(n^{\log_3 4})$.

Desta forma como $T(n) \in O(n^{\log_3 4})$ e $T(n) \in \Omega(n^{\log_3 4})$ temos que $T(n) \in \Theta(n^{\log_3 4})$.

Exercício 8

Prove pelo método mestre que a seguinte relação de recorrência

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 4T(n/2) + n^2, & n > 1 \end{cases}$$

pertence à $\Theta(n^2 \lg n)$.

Solução:

Pelo método mestre, temos que:

$n^2 \in \Theta(n^{\log_2 4} = n^2)$

Portanto $T(n) \in \Theta(n^2 \log n)$.

Exercício 9

Ache uma cota justa para a relação de recorrência:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 2T(n-1) + 1, & n > 1 \end{cases}$$

Solução: Mostraremos que:

$$T(n) \leq c2^n$$

Pelo método da substituição:

$$T(n) \leq 2c(2)^{n-1} + 1$$
$$= 2^{n+1} + 1 \gneq 2^n$$

Reforçando a hipótese de indução,mostraremos que:

$$T(n) \leq c2^n - b$$

Pelo método da substituição:

$$\begin{aligned} T(n) \leq & 2(c(2)^{n-1} - b) + 1 \\ & c2^n - 2b + 1 \leq c2^n - b \qquad \diamond \text{ com } b \geq 1 \end{aligned}$$

Assim, para qualquer $c > 0$ e n suficientemente grande, temos que $T(n) \in O(2^n)$.

Mostraremos agora que

$$T(n) \geq c2^n$$

Pelo método da substituição:

$$\begin{aligned} T(n) \geq & 2c(2)^{n-1} + 1 \\ = & 2^{n+1} + 1 \geq 2^n \end{aligned}$$

Assim, para qualquer constante $c > 0$, e n suficientemente grande, temos que $T(n) \in \Omega(2^n)$.

Como ($T(n) \in O(2^n)$ e $T(n) \in \Omega(2^n)$, temos que $T(n) \in \Theta(2^n)$.

Exercício 10

Ache uma cota justa para a relação de recorrência:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3T(\sqrt{n}) + \lg n, & n > 1 \end{cases}$$

Utilizando mudança de variáveis.

Solução: Tome $m = \log n$, isto é, $n = 2^m$. Realizando a mudança de variáveis, temos:

$$T(2^m) = \begin{cases} \Theta(1), & n = 1 \\ 3T(2^{\frac{m}{2}}) + m, & n > 1 \end{cases}$$

Tomando $T(2^m)$ como $S(m)$, temos:

$$S(m) = \begin{cases} \Theta(1), & n = 1 \\ 3S(\frac{m}{2}) + m, & n > 1 \end{cases}$$

Pelo método mestre:

$$m \in O(n^{\log_2 3 - \epsilon}) \qquad \text{, com } \epsilon < \log_2 3 - 1$$

Assim, $S(m) \in \Theta(m^{\log_2 3})$.

Realizando a mudança de variáveis, temos que:

$T(n) \in \Theta((\log n)^{\log_2 3})$

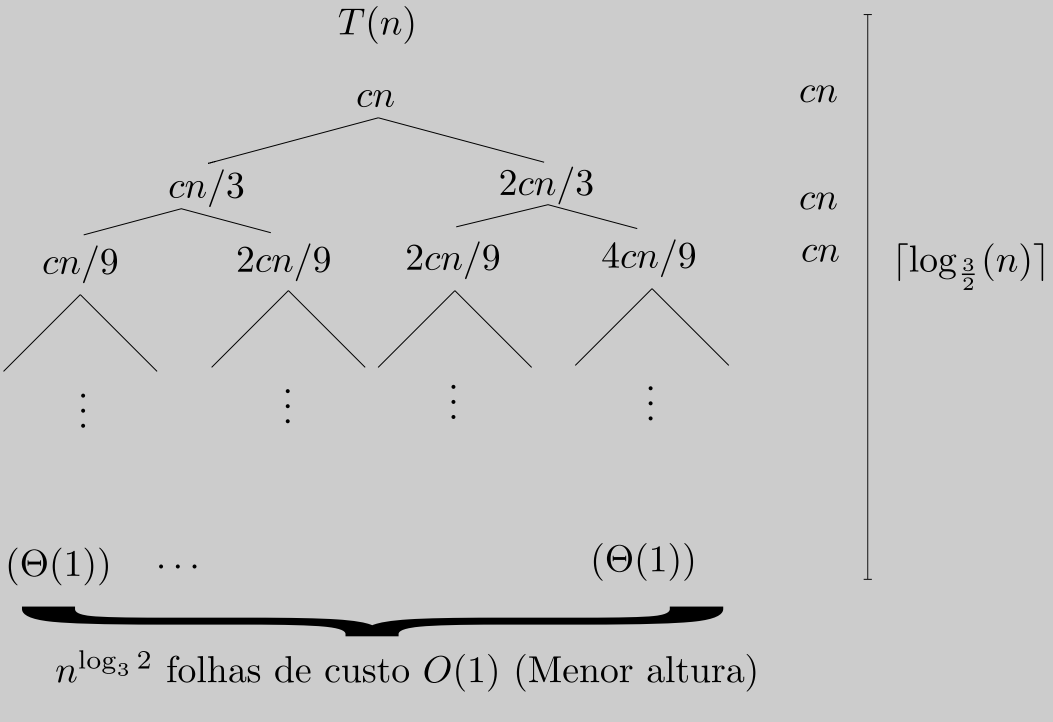
Exercício 11

Utilize a árvore de recursão para achar uma aproximação de uma cota superior e inferior para a relação de recorrência

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(n/3) + T(2n/3) + \Theta(n), & n > 1 \end{cases}$$

Prove as cotas com o método da substituição.

Solução: Obtemos a seguinte árvore de recursão:



Assim, temos de maneira aproximada (superestimando):

$$\sum_{i=0}^{\lceil \log_{\frac{3}{2}} n \rceil} cn + n$$
$$= cn \log_{\frac{3}{2}} n + n$$

Utilizando uma cota inferior de $T(n) \in \Omega(n \log n)$ tentaremos provar pelo método da substituição:

$$\begin{aligned}
T(n) \geq & c\left(\frac{n}{3}\log\left(\frac{n}{3}\right)\right) + c\left(\frac{2n}{3}\log\left(\frac{2n}{3}\right)\right) + an \\
= & c\left(\frac{n}{3}(\log(n) - \log(3))\right) + c\left(\frac{2n}{3}(\log(2n) - \log(3))\right) + an \\
= & \frac{cn}{3}\log(n) - \frac{cn}{3}\log(3) + \frac{2cn}{3}\log(2n) - \frac{2cn}{3}\log(3) + an \\
= & \frac{cn}{3}\log(n) + \frac{2cn}{3}\log(2n) - cn\log(3) + an \\
= & cn\log n + \frac{2cn}{3}\log(2) - cn\log(3) + an
\end{aligned}$$

Precisamos agora encontrar as condições de existência de c para que:

$$cn\log n + \frac{2cn}{3}\log(2) - cn\log(3) + an \geq cn\log n$$

$$\begin{aligned}
& cn\log n + \frac{2cn}{3}\log(2) - cn\log(3) + an \geq cn\log n \\
& \frac{2cn}{3}\log(2) - cn\log(3) + an \geq 0 \\
& \frac{2c}{3}\log(2) - c\log(3) + a \geq 0 \\
& c\left(\frac{2\log(2)}{3} - \log(3)\right) + a \geq 0 \\
& c \geq \frac{-a}{\left(\frac{2\log(2)}{3} - \log(3)\right)} \\
& c \geq \frac{a}{\log(3) - \frac{2\log(2)}{3}}
\end{aligned}$$

Exercício 12

Utilize o método mestre para descobrir uma cota justa para as relações de recorrência a seguir. Considere que o caso base é simples o suficiente:

(a) $T(n) = 2T(n/4) + 1$

(b) $T(n) = 2T(n/4) + \sqrt{n}$

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

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

Solução:

a)

De acordo com o método mestre:

$$1 \in O(n^{\log_4 2-\epsilon} = n^{0.5-\epsilon}), \qquad \text{com } \epsilon < 0.5$$

Logo $T(n) \in \Theta(\sqrt{n})$

b)

De acordo com o método mestre:

$$\sqrt{n} \in \Theta(n^{\log_4 2} = n^{0.5})$$

Logo $T(n) \in \Theta(\sqrt{n} \log n)$ c)

$$n \in \Omega(n^{\log_4 2+\epsilon} = n^{0.5+\epsilon}), \qquad \text{com } \epsilon < 0.5$$

A condição de regularidade é respeitada, uma vez que:

$$\frac{2n}{4} \leq cn, \qquad \text{com } 0.5 \leq c < 1 \text{ e } n \text{ suficientemente grande}$$

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

$$n^2 \in \Omega(n^{\log_4 2+\epsilon} = n^{0.5+\epsilon}), \qquad \text{com } \epsilon < 1.5$$

A condição de regularidade é respeitada, uma vez que:

$$\frac{4n^2}{16} \leq cn^2, \qquad \text{com } \frac{1}{4} \leq c < 1 \text{ e } n \text{ suficientemente grande}$$

Logo $T(n) \in \Theta(n^2)$

Exercício 13

O método mestre pode ser aplicado para a relação de recorrência a seguir?

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 4T(n/2) + n^2 \lg n, & n > 1 \end{cases}$$

Dê uma cota superior justa para $T(n)$.

Solução:

O método mestre não é aplicável, visto que $n^2 \lg n$ não cresce polinomialmente mais do que $n^{\log_2 4} = n^2$.

Mostraremos pelo método da substituição que:

$$T(n) \in O(n^2 \lg^2 n)$$

Aplicando-o:

$$\begin{aligned}
T(n) \leq & 4c(\frac{n}{2}^2 \lg(\frac{n}{2})^2) + n^2 \lg n \\
& = cn^2(\lg(n) - 1)^2 + n^2 \\
& = cn^2 \lg^2(n) - 2cn^2 \lg(n) + cn^2 + n^2 \lg n
\end{aligned}$$

Precisamos mostrar a condição de existência de c para que:

$$cn^2 \lg^2(n) - 2cn^2 \lg(n) + cn^2 + n^2 \lg n \leq cn^2 \lg^2(n)$$

$$\begin{aligned}
& cn^2 \lg^2 n - 2cn^2 \lg n + cn^2 + n^2 \lg n \leq cn^2 \lg^2(n) \\
& -2cn^2 \lg n + cn^2 + n^2 \lg n \leq 0 \\
& -2c \lg n + c + \lg n \leq 0 \\
& c(-2 \lg n + 1) \leq -\lg n \\
& c(2 \lg n + 1) \geq \lg n \\
& c \geq \frac{\lg n}{2 \lg(n) + 1} \\
& c \geq \frac{1}{3} \qquad , \text{ a partir de } n > 2
\end{aligned}$$

Assim, com $c \geq \frac{1}{3}$, e com n suficientemente grande, temos $T(n) \in O(n^2 \log^2 n)$