Resolvendo

Lista 2

Universidade Federal de Minas Gerais
Departamento de Computação
Projeto e Análise de Algoritmos - 2024.2
Professor: Marcio Costa Santos
Lista 2

Considere que todas as recorrência descritas possuem caso base (ou casos bases) iguais a 1.

Exercício 1. Determine e prove uma equivalência assintótica para todas as recorrências abaixo

1.1. $T(n) = T(n-3) + 1$ || R: $\Theta(n)$

- **Exemplificando alguns passos da recorrência**
  - $T(n) = T(n-3) + 1$
  - $T(n-3) = T(n-6) + 1$
  - $T(n-6) = T(n-9) + 1$
  - $\vdots$
- **Descendo alguns níveis da recorrência**
  - $T(n) = T(n-3) + 1$
  - $T(n) = T(n-6) + 2$
  - $T(n) = T(n-9) + 3$
  - $\vdots$
  - $T(n) = T(n-k*3) + k$
- **Encontrando o caso base**
  - $n - k*3 = 0$
  - $n = k*3$
  - $n = 3k$
  - $k = \frac{n}{3}$
- **Retornando aos níveis da recorrência**
  - $T(n) = T(n-k*3) + k$
  - $T(n) = T(n-\frac{n}{3}*3) + \frac{n}{3}$
  - $T(n) = T(n-n) + \frac{n}{3}$
  - $T(n) = T(0) + \frac{n}{3}$
  - $\Theta(T(n)) = \Theta(T(0) + \frac{n}{3})$
  - $\Theta(T(n)) = \Theta(T(0)) + \Theta(\frac{n}{3})$
    - Por definição: $\Theta(T(0)) = \Theta(1)$
  - $\Theta(T(n)) = \Theta(1) + \Theta(\frac{n}{3})$
  - $\Theta(T(n)) = \Theta(n)$

1.2. $T(n) = 2T(n-2) + \log n$ || R: $\Theta(2^{n/2})$

- $T(n) = 2T(n-2) + \log(n)$
  - $T(n-2) = 2T(n-4) + \log(n-2)$
- $T(n) = 2(2T(n-4) + \log(n-2)) + \log(n)$
  - $T(n-4) = 2T(n-6) + \log(n-4)$
- $T(n) = 2(2(2T(n-6) + \log(n-4)) + \log(n-2)) + \log(n)$
  - $T(n-6) = 2T(n-8) + \log(n-6)$
- $T(n) = 2(2(2(2T(n-8) + \log(n-6)) + \log(n-4)) + \log(n-2)) + \log(n)$
- **Distribuindo**
- $T(n) = (((2*2*2*2T(n-8) + 2*2*2*\log(n-6)) + 2*2*\log(n-4)) + 2*\log(n-2)) + \log(n)$
- $T(n) = 2^4T(n-8) + 2^3\log(n-6) + 2^2\log(n-4) + 2^1\log(n-2) + 2^0\log(n)$
- **Generalizando**
- $T(k) = 2^kT(n-2k) + \sum_{i=0}^{k-1} 2^i \log(n-2i)$
  - **Substituindo k para alcançar o caso base**

- $n - 2k = 0$
- $n = 2k$
- $k = n/2$
- $T(n/2) = 2^{n/2}T(n - 2n/2) + \sum_{i=0}^{(n/2)-1} 2^i \log(n - 2i)$
- **Separando para simplificar**
- $T(n/2) = X + Y$
  - $X = 2^{n/2}T(0)$
    - $\Theta(X) = \Theta(2^{n/2})$
  - $Y = \sum_{i=0}^{(n/2)-1} 2^i \log(n - 2i)$
    - $\Theta(Y) = \Theta(\sum_{i=0}^{(n/2)-1} 2^i \log(n - 2i))$
    - $\Theta(Y) = \Theta((2^{n/2} - 1)\log(n - 2(n/2) - 1))$
    - $\Theta(Y) = \Theta((2^{n/2} - 1)\log(n - n - 1))$
    - $\Theta(Y) = \Theta((2^{n/2} - 1)\log(-1))$ [Esse log negativo tá esquisito...]
- **Como $\Theta(X) > \Theta(Y)$...**
- $\Theta(T(n)) = \Theta(2^{n/2})$

1.3. $T(n) = T(n-1) + n$ || R: $\Theta(n^2)$

- $P_0 : T(n) = T(n-1) + n$
- $P_1 : T(n-1) = T(n-2) + (n-1)$
- $P_2 : T(n-2) = T(n-3) + (n-2)$
- $P_3 : T(n-3) = T(n-4) + (n-3)$
- $\vdots$

Substituindo os passos no caso base:

- $T(n) = T(n-1) + n$
- $T(n) = (T(n-2) + (n-1)) + n$
  - $T(n) = T(n-2) + (n-1) + n$
  - $T(n) = T(n-2) + 2n - 1$
- $T(n) = T(n-3) + (n-2) + 2n - 1$
  - $T(n) = T(n-3) + 3n - 3$
- $T(n) = T(n-3) + 3n - 3$
  - $T(n) = T(n-4) + (n-3) + 3n - 3$
  - $T(n) = T(n-4) + 4n - 6$
- $\vdots$
- $T(n) = T(n-k) + kn - \sum_{i=0}^{k-1} i$
- $T(n) = T(n-k) + kn - \frac{k \cdot (k-1)}{2}$

Calculando k para alcançar o caso base:

- $T(n-k) = T(0)$
- $n - k = 0$
- $n = k$

Substituindo k na equação:

- $T(n) = T(0) + n^2 - \frac{n \cdot (n-1)}{2}$
- $T(n) = 1 + n^2 - \frac{n^2 - n}{2}$
- $T(n) = 1 + n^2 + \frac{-n^2 + n}{2}$
- $T(n) = 1 + n^2 + \frac{-n^2}{2} + \frac{n}{2}$
- $T(n) = \Theta(1) + \Theta(n^2) + \Theta(\frac{-n^2}{2}) + \Theta(\frac{n}{2})$
- $T(n) = \Theta(n^2)$

1.4. $T(n) = 2T(n-1) + n^2 + 1$ || $\Theta(2^n * n^2)$

- **Exemplificando alguns passos da recorrência**
  - $T(n) = 2^1 * (T(n-1)) + n^2 + 1$
  - $T(n-1) = 2^1 * (T(n-2)) + n^2 + 1$
  - $T(n-2) = 2^1 * (T(n-3)) + n^2 + 1$
  - $\vdots$
- **Descendo alguns níveis da recorrência**
  - $T(n) = 2^1 * (T(n-1)) + n^2 + 1$

- $T(n) = 2^1 * (2^1 * (T(n-2)) + n^2 + 1) + 1$
- $T(n) = 2^1 * (2^1 * (2^1 * (T(n-3)) + n^2 + 1) + n^2 + 1) + n^2 + 1$

- **Simplificando**
  - $T(n) = 2^1 * 2^1 * 2^1 * (T(n-3)) + 2^1 * 2^1 * n^2 + 2^1 * 2^1 * 1 + 2^1 * n^2 + 2^1 * 1 + n^2 + 1$
  - $T(n) = 2^3 * (T(n-3)) + 2^2 * n^2 + 2^2 + 2^1 * n^2 + 2^1 + n^2 + 1$
  - **Separando os blocos**
    - $T(n) = [2^3 * (T(n-3))] + [2^2 * n^2 + 2^1 * n^2 + 2^0 * n^2] + [2^2 + 2^1 + 2^0]$
  - **Generalizando**
    - $T(n) = 2^k * (T(n-k)) + \sum_{i=0}^{k-1} 2^i * n^2 + \sum_{i=0}^{k-1} 2^i$
- **Encontrando o caso base**
  - $n - k = 0$
  - $n = k$
- **Retornando aos níveis da recorrência**
  - $T(n) = 2^k * (T(n-k)) + \sum_{i=0}^{k-1} 2^i * n^2 + \sum_{i=0}^{k-1} 2^i$
  - **Substituindo $n = k$**
    - $T(n) = 2^n * (T(n-n)) + \sum_{i=0}^{n-1} 2^i * n^2 + \sum_{i=0}^{n-1} 2^i$
    - $T(n) = 2^n * (T(0)) + \sum_{i=0}^{n-1} 2^i * n^2 + \sum_{i=0}^{n-1} 2^i$
    - **Separando em blocos**
      - $T(n) = A + B + C$
        - $A = 2^n * (T(0))$
          - $\Theta(A) = \Theta(2^n)$
        - $B = \sum_{i=0}^{n-1} 2^i * n^2$
          - $\Theta(B) = \Theta(\sum_{i=0}^{n-1} 2^i * n^2)$
          - $\Theta(B) = \Theta(2^{n-1} * n^2)$
          - $\Theta(B) = \Theta(2^n * n^2)$
        - $C = \sum_{i=0}^{n-1} 2^i$
          - $\Theta(C) = \Theta(\sum_{i=0}^{n-1} 2^i)$
          - $\Theta(C) = \Theta(2^{n-1})$
          - $\Theta(C) = \Theta(2^n)$
      - $\Theta(T(n)) = \Theta(A + B + C)$
      - $\Theta(T(n)) = \Theta(A) + \Theta(B) + \Theta(C)$
      - $\Theta(T(n)) = \Theta(2^n) + \Theta(2^n * n^2) + \Theta(2^n)$
      - $\Theta(T(n)) = \Theta(2^n * n^2)$

Exercício 2. Determine e prove uma equivalência assintótica para todas as recorrências abaixo. Não use o teorema mestre

2.1. $T(n) = 2T(\frac{n}{2}) + 1$ || R: $\Theta(n)$

- **Exemplificando alguns passos da recorrência**
  - $T(n * 2^0) = 2^1 * (T(n * 2^{-1})) + 1$
  - $T(n * 2^{-1}) = 2^1 * (T(n * 2^{-2})) + 1$
  - $T(n * 2^{-2}) = 2^1 * (T(n * 2^{-3})) + 1$
  - $T(n * 2^{-3}) = 2^1 * (T(n * 2^{-4})) + 1$
  - $\vdots$
- **Descendo alguns níveis da recorrência**
  - $T(n) = 2^1 * (T(n * 2^{-1})) + 1$
  - $T(n) = 2^1 * (2^1 * (T(n * 2^{-2})) + 1) + 1$
  - $T(n) = 2^1 * (2^1 * (2^1 * (T(n * 2^{-3})) + 1) + 1) + 1$
  - $T(n) = 2^1 * (2^1 * (2^1 * (2^1 * (T(n * 2^{-4})) + 1) + 1) + 1) + 1$
- **Simplificando**
  - $T(n) = 2^4 * (T(n * 2^{-4})) + 2^3 + 2^2 + 2^1 + 2^0$
  - **Separando os blocos**
    - $T(n) = 2^4 * (T(n * 2^{-4})) + [2^3 + 2^2 + 2^1 + 2^0]$
  - **Generalizando**
    - $T(n) = 2^k * (T(n * 2^{-k})) + \sum_{i=0}^{k} 2^i$
- **Encontrando o caso base**
  - $n * 2^{-k} = 1$
  - $n/2^k = 1$
  - $n = 2^k$
  - $k = \log_2 n$
- **Retornando aos níveis da recorrência**

    - $T(n) = 2^k * (T(n * 2^{-k})) + \sum_{i=0}^{k} 2^i$
    - **Substituindo $k = \log_2 n$**
        - $T(n) = 2^{\log_2 n} * (T(n * 2^{-\log_2 n})) + \sum_{i=0}^{\log_2 n} 2^i$
        - $T(n) = n * (T(1)) + \sum_{i=0}^{\log_2 n} 2^i$
        - **Separando em blocos**
            - $T(n) = A + B$
                - $A = n * (T(1))$
                    - $\Theta(A) = \Theta(n)$
                - $B = \sum_{i=0}^{\log_2 n} 2^i$
                    - $\Theta(B) = \Theta(\sum_{i=0}^{\log_2 n} 2^i)$
                    - $\Theta(B) = \Theta(2^{\log_2 n})$
                    - $\Theta(B) = \Theta(n)$
            - $\Theta(T(n)) = \Theta(A + B) = \Theta(A) + \Theta(B)$
            - $\Theta(T(n)) = \Theta(n) + \Theta(n)$
            - $\Theta(T(n)) = \Theta(n)$

2.2. $T(n) = 4T(\frac{n}{2}) + \log n$ || R: $\Theta(n^2 * \log n)$

- **Exemplificando alguns passos da recorrência**
    - $T(n * 2^0) = 4 * (T(n * 2^{-1})) + \log(n * 2^0)$
    - $T(n * 2^{-1}) = 4 * (T(n * 2^{-2})) + \log(n * 2^{-1})$
    - $T(n * 2^{-2}) = 4 * (T(n * 2^{-3})) + \log(n * 2^{-2})$
    - $T(n * 2^{-3}) = 4 * (T(n * 2^{-4})) + \log(n * 2^{-3})$
    - $\vdots$
- **Descendo alguns níveis da recorrência**
    - $T(n) = 4 * (T(n * 2^{-1})) + \log(n * 2^0)$
    - $T(n) = 4 * (4 * (T(n * 2^{-2})) + \log(n * 2^{-1})) + \log(n * 2^0)$
    - $T(n) = 4 * (4 * (4 * (T(n * 2^{-3})) + \log(n * 2^{-2})) + \log(n * 2^{-1})) + \log(n * 2^0)$
    - $T(n) = 4 * (4 * (4 * (4 * (T(n * 2^{-4})) + \log(n * 2^{-3})) + \log(n * 2^{-2})) + \log(n * 2^{-1})) + \log(n * 2^0)$
- **Simplificando**
    - $T(n) = 4^4 * (T(n * 2^{-4})) + 4^3 * \log(n * 2^{-3}) + 4^2 * \log(n * 2^{-2}) + 4^1 * \log(n * 2^{-1}) + 4^0 * \log(n * 2^0)$
    - **Separando os blocos**
        - $T(n) = 4^4 * (T(n * 2^{-4})) + [4^3 * \log(n * 2^{-3}) + 4^2 * \log(n * 2^{-2}) + 4^1 * \log(n * 2^{-1}) + 4^0 * \log(n * 2^0)]$
        - $T(n) = 4^4 * (T(n * 2^{-4})) + [4^3 * (\log(n) + \log(2^{-3})) + 4^2 * (\log(n) + \log(2^{-2})) + 4^1 * (\log(n) + \log(2^{-1})) + 4^0 * (\log(n) + \log(2^0))]$
        - $T(n) = 4^4 * (T(n * 2^{-4})) + [4^3 * \log(n) + 4^3 * \log(2^{-3}) + 4^2 * \log(n) + 4^2 * \log(2^{-2}) + 4^1 * \log(n) + 4^1 * \log(2^{-1}) + 4^0 * \log(n) + 4^0 * \log(2^0)]$
        - $T(n) = 4^4 * (T(n * 2^{-4})) + [\log(n) * (4^3 + 4^2 + 4^1 + 4^0) + \log(2^{-3}) * 4^3 + \log(2^{-2}) * 4^2 + \log(2^{-1}) * 4^1 + \log(2^0) * 4^0]$
    - **Generalizando**
        - $T(n) = 4^k * (T(n * 2^{-k})) + \log(n) * \sum_{i=0}^{k} 4^i + \sum_{i=0}^{k} \log(2^{-i}) * 4^i$ [JV: Esse log não deveria estar negativo...]
- **Encontrando o caso base**
    - $n * 2^{-k} = 1$
    - $n/2^k = 1$
    - $n = 2^k$
    - $k = \log_2 n$
- **Retornando aos níveis da recorrência**
    - $T(n) = 4^k * (T(n * 2^{-k})) + \log(n) * \sum_{i=0}^{k} 4^i + \sum_{i=0}^{k} \log(2^{-i}) * 4^i$
    - **Substituindo $k = \log_2 n$**
        - $T(n) = 4^{\log_2 n} * (T(n * 2^{-\log_2 n})) + \log(n) * \sum_{i=0}^{\log_2 n} 4^i + \sum_{i=0}^{\log_2 n} \log(2^{-i}) * 4^i$
        - $T(n) = [n^2 * (T(1))] + [\log(n) * \sum_{i=0}^{\log_2 n} 4^i] + [\sum_{i=0}^{\log_2 n} \log(2^{-i}) * 4^i]$
        - **Separando em blocos**
            - $T(n) = A + B + C$
                - $A = n^2 * (T(1))$
                    - $\Theta(A) = \Theta(n^2)$
                - $B = \log(n) * \sum_{i=0}^{\log_2 n} 4^i$
                    - $\Theta(B) = \Theta(\log(n) * \sum_{i=0}^{\log_2 n} 4^i)$
                    - $\Theta(B) = \Theta(\log(n) * 4^{\log_2 n})$
                    - $\Theta(B) = \Theta(\log(n) * n^2)$
                    - $\Theta(B) = \Theta(n^2 * \log(n))$
                - $C = \sum_{i=0}^{\log_2 n} \log(2^{-i}) * 4^i$
                    - $\Theta(C) = \Theta(\log(2^{-\log_2 n}) * 4^{\log_2 n})$
                    - $\Theta(C) = \Theta(-\log(n) * n^2)$ [JV: vou fingir que isso é positivo e seguir daí]
                    - $\Theta(C) = \Theta(n^2 * \log(n))$
            - $\Theta(T(n)) = \Theta(A + B + C) = \Theta(A) + \Theta(B) + \Theta(C)$
            - $\Theta(T(n)) = \Theta(n^2) + \Theta(n^2 * \log(n)) + \Theta(n^2 * \log(n))$

- $\Theta(T(n)) = \Theta(n^2 * \log(n))$

2.3. $T(n) = 7T(\frac{n}{3}) + n$ || R: $\Theta(n^{\log_3 7})$

- **Exemplificando alguns passos da recorrência**
  - $T(n * 3^0) = 7 * (T(n * 3^{-1})) + n$
  - $T(n * 3^{-1}) = 7 * (T(n * 3^{-2})) + n$
  - $T(n * 3^{-2}) = 7 * (T(n * 3^{-3})) + n$
  - $T(n * 3^{-3}) = 7 * (T(n * 3^{-4})) + n$
  - $\vdots$
- **Descendo alguns níveis da recorrência**
  - $T(n) = 7 * (T(n * 3^{-1})) + n$
  - $T(n) = 7 * (7 * (T(n * 3^{-2})) + n) + n$
  - $T(n) = 7 * (7 * (7 * (T(n * 3^{-3})) + n) + n) + n$
  - $T(n) = 7 * (7 * (7 * (7 * (T(n * 3^{-4})) + n) + n) + n) + n$
- **Simplificando**
  - $T(n) = 7^4 * (T(n * 3^{-4})) + 7^3 * n + 7^2 * n + 7^1 * n + 7^0 * n$
  - **Separando os blocos**
    - $T(n) = 7^4 * (T(n * 3^{-4})) + [7^3 * n + 7^2 * n + 7^1 * n + 7^0 * n]$
  - **Generalizando**
    - $T(n) = 7^k * (T(n * 3^{-k})) + \sum_{i=0}^{k} 7^i * n$
    - $T(n) = 7^k * (T(n * 3^{-k})) + n * \sum_{i=0}^{k} 7^i$
- **Encontrando o caso base**
  - $n * 3^{-k} = 1$
  - $n/3^k = 1$
  - $n = 3^k$
  - $k = \log_3 n$
- **Retornando aos níveis da recorrência**
  - $T(n) = 7^k * (T(n * 3^{-k})) + n * \sum_{i=0}^{k} 7^i$
  - **Substituindo** $k = \log_3 n$
    - $T(n) = 7^{\log_3 n} * (T(n * 3^{-\log_3 n})) + n * \sum_{i=0}^{\log_3 n} 7^i$
    - $T(n) = n^{\log_3 7} * (T(1)) + n * \sum_{i=0}^{\log_3 n} 7^i$
    - **Separando em blocos**
      - $T(n) = A + B$
        - $A = n^{\log_3 7} * (T(1))$
          - $\Theta(A) = \Theta(n^{\log_3 7})$
        - $B = n * \sum_{i=0}^{\log_3 n} 7^i$
          - $\Theta(B) = \Theta(n * \sum_{i=0}^{\log_3 n} 7^i)$
          - $\Theta(B) = \Theta(n * 7^{\log_3 n})$
          - $\Theta(B) = \Theta(n * n^{\log_3 7})$
          - $\Theta(B) = \Theta(n^{\log_3 7 + 1})$
          - $\Theta(B) = \Theta(n^{\log_3 7})$
      - $\Theta(T(n)) = \Theta(A + B) = \Theta(A) + \Theta(B)$
      - $\Theta(T(n)) = \Theta(n^{\log_3 7}) + \Theta(n^{\log_3 7})$
      - $\Theta(T(n)) = \Theta(n^{\log_3 7})$

Exercício 3. Usando o teorema mestre determine uma equivalência assintótica para

- Teorema Mestre
  - Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função, e $T(n) = aT(\frac{n}{b}) + f(n)$, então, para algum $\epsilon > 0$:
    - Se $f(n) = O(n^{\log_b(a) - \epsilon}) \implies T(n) = \Theta(n^{\log_b(a)})$ [<=]
    - Se $f(n) = \Theta(n^{\log_b(a)}) \implies T(n) = \Theta(n^{\log_b(a)} * log(n))$ [=]
    - Se $f(n) = \Omega(n^{\log_b(a) + \epsilon})$ e $af(\frac{n}{b}) \leq cf(n)$ então $\implies T(n) = \Theta(f(n))$ [>=]

$$\log_a b = x \Leftrightarrow a^x = b$$

- $a =$ Base
- $b =$ Logaritmando
- $x =$ Logaritmo

3.1. $T(n) = 2T(\frac{n}{4}) + 1$ || $T(n) = \Theta(\sqrt{n})$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 4; f(n) = 1$
- **Depois calculamos** $\log_b(a)$
  - $\log_4(2) = \frac{1}{2}$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a)\pm\epsilon} = n^{\log_4(2)\pm\epsilon} = n^{\frac{1}{2}\pm\epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se** $f(n) = O(n^{\log_b(a)-\epsilon})$
    - $f(n) = 1 = O(n^{\frac{1}{2}-\epsilon})$
    - **Agora, deve-se buscar um** $\epsilon > 0$ **que satisfaça a equação**
      - $1 = n^0$
      - Igualando os expoentes, temos:
      - $0 \leq \frac{1}{2} - \epsilon$
      - $\epsilon \leq \frac{1}{2}$
    - $1 = O(n^{\frac{1}{2}-\frac{1}{2}}) = O(n^0) = O(1)$
    - $1 = O(1)$
    - **Com isso, o CASO 1 é verdadeiro, o que implica em:** $T(n) = \Theta(n^{\log_b(a)})$
      - $T(n) = \Theta(n^{\log_4(2)}) = \Theta(n^{\frac{1}{2}}) = \Theta(\sqrt{n})$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(\sqrt{n})$

3.2. $T(n) = 2T(\frac{n}{4}) + n$ || $T(n) = \Theta(n)$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 4; f(n) = n$
- **Depois calculamos** $\log_b(a)$
  - $\log_4(2) = \frac{1}{2}$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a)\pm\epsilon} = n^{\log_4(2)\pm\epsilon} = n^{\frac{1}{2}\pm\epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se** $f(n) = O(n^{\log_b(a)-\epsilon})$
    - $f(n) = n = O(n^{\frac{1}{2}-\epsilon})$
    - **Agora, deve-se buscar um** $\epsilon > 0$ **que satisfaça a equação**
      - $n^1 \leq n^{\frac{1}{2}-\epsilon}$
      - Igualando os expoentes, temos:
      - $1 \leq \frac{1}{2} - \epsilon$
      - $\epsilon \leq \frac{1}{2} - 1$
      - $\epsilon \leq \frac{-1}{2}$ [JV: Não pode ser negativo]
    - **Como** ϵ **não pode ser negativo, o CASO 1 é falso**
  - **CASO 2: verificar se** $f(n) = \Theta(n^{\log_b(a)})$
    - $f(n) = n = \Theta(n^{\frac{1}{2}})$
    - $n = n^{\frac{1}{2}}$
    - $n = \sqrt{n}$ [JV: Não é verdade]
    - **Como a igualdade não é verdadeira, o CASO 2 é falso**
  - **CASO 3: verificar se** $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ **e** $af(\frac{n}{b}) \leq cf(n)$
    - **Primeiro, verifica-se** $f(n) = \Omega(n^{\log_b(a)+\epsilon})$
      - $f(n) = n = \Omega(n^{\frac{1}{2}+\epsilon})$
      - **Agora, deve-se buscar um** $\epsilon > 0$ **que satisfaça a equação**
        - $n^1 \geq n^{\frac{1}{2}+\epsilon}$
        - $1 \geq \frac{1}{2} + \epsilon$
        - $\epsilon \leq 1 - \frac{1}{2}$
        - $\epsilon \leq \frac{1}{2}$
        - Sendo assim, $\epsilon \in]0, \frac{1}{2}]$
        - Escolhendo um valor para $\epsilon = \frac{1}{4}$
        - $n = \Omega(n^{\frac{1}{2}+\frac{1}{4}})$
        - $n = \Omega(n^{\frac{3}{4}})$
        - $n = \Omega(n^{0.75})$
        - **Como a igualdade é verdadeira, a primeira parte do CASO 3 é verdadeiro**
    - **Agora, verifica-se** $af(\frac{n}{b}) \leq cf(n)$
      - $2 * f(\frac{n}{4}) \leq c * f(n)$
      - $2 * \frac{n}{4} \leq c * n$

- $\frac{n}{2} \leq c * n$
- **Se considerarmos $c = \frac{1}{2}$, temos que:**
  - $\frac{n}{2} \leq \frac{1}{2} * n$
  - $\frac{n}{2} \leq \frac{n}{2}$
- **Com isso, o CASO 3 é verdadeiro, o que implica em:** $T(n) = \Theta(f(n))$
  - $T(n) = \Theta(n)$

- **Com isso, conclui-se que, pelo CASO 3 do Teorema Mestre:**
  - $T(n) = \Theta(n)$

3.3. $T(n) = 2T(\frac{n}{4}) + \log n$ || $T(n) = \Theta(\sqrt{n})$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 4; f(n) = \log n$
- **Depois calculamos $\log_b(a)$**
  - $\log_4(2) = \frac{1}{2}$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a) \pm \epsilon} = n^{\log_4(2) \pm \epsilon} = n^{\frac{1}{2} \pm \epsilon}$
  - $n^{\frac{1}{2} \pm \epsilon}$
  - **Agora testaremos cada caso**
    - **CASO 1: verifica-se $f(n) = O(n^{\log_b(a) - \epsilon})$**
      - $f(n) = \log n = O(n^{\frac{1}{2} - \epsilon})$
      - **Agora, deve-se buscar um $\epsilon > 0$ que satisfaça a equação**
        - Testando $\epsilon = \frac{1}{4}$
          - $n^{\frac{1}{2} - \frac{1}{4}} = n^{\frac{1}{4}}$
        - $f(n) = \log n = O(n^{\frac{1}{4}})$
        - Como $\log n$ tende a ser menor que $\sqrt[4]{n}$
      - **Com isso, o CASO 1 é verdadeiro, o que implica em:** $T(n) = \Theta(n^{\log_b(a)})$
        - $T(n) = \Theta(n^{\log_4(2)}) = \Theta(n^{\frac{1}{2}})$
        - $T(n) = \Theta(\sqrt{n})$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(\sqrt{n})$

3.4. $T(n) = 4T\left(\frac{n}{2}\right) + 1$ || $T(n) = \Theta(n^2)$

- **Primeiro definimos os 3 termos principais**
  - $a = 4; b = 2; f(n) = 1$
- **Depois calculamos $\log_b(a)$**
  - $\log_2(4) = 2$
- **Então substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a) \pm \epsilon} = n^{2 \pm \epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se $f(n) = O\left(n^{\log_b(a) - \epsilon}\right)$**
    - Temos $f(n) = 1$ e queremos verificar se existe algum $\epsilon > 0$ tal que:

$$1 = O\left(n^{2-\epsilon}\right)$$

    - Escolhendo, por exemplo, $\epsilon = 1$, temos $n^{2-1} = n$. Como $1 = O(n)$ (pois para $n \geq 1$, $1 \leq c \cdot n$ para alguma constante c), a condição é satisfeita.
    - **Portanto, o CASO 1 é verdadeiro.**
  - **Consequentemente, pelo CASO 1 do Teorema Mestre, concluímos que:**
    - $T(n) = \Theta\left(n^{\log_b(a)}\right) = \Theta(n^2)$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(n^2)$

3.5. $T(n) = 4T(n/2) + n$ || $T(n) = \Theta(n^2)$

- **Primeiro definimos os 3 termos principais**
  - $a = 4; b = 2; f(n) = n$
- **Depois calculamos $\log_b(a)$**
  - $\log_2(4) = 2$
- **Então substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a) \pm \epsilon} = n^{2 \pm \epsilon}$

- **Agora testaremos cada caso**
  - **CASO 1: verificar se** $f(n) = O(n^{\log_b(a)-\epsilon})$
    - $n = O(n^{2-\epsilon})$
    - Se escolhermos, por exemplo, $\epsilon = 1$, então $n^{2-1} = n$.
    - Assim, temos $n = O(n)$, o que é verdadeiro.
    - **Portanto, o CASO 1 é satisfeito.**
- **Consequentemente, pelo CASO 1 do Teorema Mestre, concluímos que:**
  - $T(n) = \Theta(n^{\log_b(a)}) = \Theta(n^2)$
- **Com isso, conclui-se que, pelo Teorema Mestre:**
  - $T(n) = \Theta(n^2)$

3.6. $T(n) = 4T(n/2) + \log n$

- **Primeiro definimos os 3 termos principais**
  - $a = 4; b = 2; f(n) = \log n$
- **Depois calculamos** $\log_b(a)$
  - $\log_2(4) = 2$
- **Então substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a)\pm\epsilon} = n^{2\pm\epsilon}$
- **Agora testaremos os casos do Teorema Mestre, na ordem correta:**
  - **CASO 1: verificar se** $f(n) = O(n^{\log_b(a)-\epsilon})$
    - Precisamos encontrar um $\epsilon > 0$ tal que:
      - $\log n = O(n^{2-\epsilon})$
    - Como $\log n$ cresce mais lentamente que qualquer potência positiva de n, podemos escolher $\epsilon = 1$, o que dá:
      - $\log n = O(n^1)$, que é verdadeiro.
    - **Com isso, o CASO 1 é verdadeiro, o que implica em:** $T(n) = \Theta(n^2)$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(n^2)$

3.7. $T(n) = 2T\left(\frac{n}{2}\right) + 1$ || $T(n) = \Theta(n)$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 2; f(n) = 1$
- **Depois calculamos** $\log_b(a)$
  - $\log_2(2) = 1$
- **Então substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a)\pm\epsilon} = n^{1\pm\epsilon}$
- **Agora testaremos os casos do Teorema Mestre, na ordem correta:**
  - **CASO 1: verificar se** $f(n) = O(n^{\log_b(a)-\epsilon})$
    - Precisamos encontrar um $\epsilon > 0$ tal que:
      - $1 = O(n^{1-\epsilon})$
    - Como qualquer potência positiva de n cresce mais rápido que 1, essa relação é verdadeira para qualquer $\epsilon > 0$.
    - **Com isso, o CASO 1 é verdadeiro, o que implica em:** $T(n) = \Theta(n)$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(n)$

3.8. $T(n) = 2T(\frac{n}{2}) + \log n$ || $T(n) = \Theta(\log n)$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 2; f(n) = \log n$
- **Depois calculamos** $\log_b(a)$
  - $\log_2(2) = 1$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a)\pm\epsilon} = n^{1\pm\epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se** $f(n) = O(n^{\log_b(a)-\epsilon})$
    - $f(n) = \log n = O(n^{1-\epsilon})$
    - **Agora, deve-se buscar um** $\epsilon > 0$ **que satisfaça a equação**
      - $\log n \leq n^{1-\epsilon}$
      - Como $\log n$ cresce mais lentamente do que qualquer potência de n, a condição é satisfeita.
    - **Como a igualdade é verdadeira, o CASO 1 é verdadeiro**
  - **Conclusão**: Pelo CASO 1 do Teorema Mestre, temos que $T(n) = \Theta(n^{\log_b(a)}) = \Theta(\log n)$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(\log n)$

3.9. $T(n) = 2T(\frac{n}{2}) + n$ || $T(n) = \Theta(n)$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 2; f(n) = n$
- **Depois calculamos $\log_b(a)$**
  - $\log_2(2) = 1$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a) \pm \epsilon} = n^{1 \pm \epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se $f(n) = O(n^{\log_b(a) - \epsilon})$**
    - $f(n) = n = O(n^{1-\epsilon})$
    - **Agora, deve-se buscar um $\epsilon > 0$ que satisfaça a equação**
      - $n^1 \leq n^{1-\epsilon}$
      - Igualando os expoentes, temos:
      - $1 \leq 1 - \epsilon$
      - $\epsilon \leq 0$ [falso, visto que $\epsilon > 0$]
    - **Como não encontramos um valor válido de ϵ que satisfaça a desigualdade, o CASO 1 é falso**
  - **CASO 2: verificar se $f(n) = \Theta(n^{\log_b(a)})$**
    - $f(n) = n = \Theta(n^1)$
    - $n = n$
    - **Como a igualdade é verdadeira, o CASO 2 é verdadeiro**
      - **Com isso, o CASO 2 é verdadeiro, o que implica em:** $T(n) = \Theta(n^{\log_b(a)})$
        - $T(n) = \Theta(n^1) = \Theta(n)$
- **Com isso, conclui-se que, pelo CASO 2 do Teorema Mestre:**
  - $T(n) = \Theta(n)$

3.10. $T(n) = 2T(\frac{n}{3}) + 1$ || $T(n) = \Theta(n^{\log_3(2)})$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 3; f(n) = 1$
- **Depois calculamos $\log_b(a)$**
  - $\log_3(2) \approx 0.631$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a) \pm \epsilon} = n^{0.631 \pm \epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se $f(n) = O(n^{\log_b(a) - \epsilon})$**
    - $f(n) = 1 = O(n^{0.631-\epsilon})$
    - **Agora, deve-se buscar um $\epsilon > 0$ que satisfaça a equação**
      - $1 \leq n^{0.631-\epsilon}$
      - Para qualquer valor de $\epsilon > 0$, sempre teremos $n^{0.631-\epsilon}$ maior que 1 à medida que n cresce.
    - **Como $\epsilon > 0$ gera uma desigualdade verdadeira, o CASO 1 é verdadeiro**
    - **Com isso, o CASO 1 é verdadeiro, o que implica em:** $T(n) = \Theta(n^{\log_b(a)})$
      - $T(n) = \Theta(n^{\log_3(2)})$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(n^{\log_3(2)})$

3.11. $T(n) = 2T(\frac{n}{3}) + \log n$ || $T(n) = \Theta(n^{\log_3(2)})$

- **Primeiro definimos os 3 termos principais**
  - $a = 2; b = 3; f(n) = \log n$
- **Depois calculamos $\log_b(a)$**
  - $\log_3(2) \approx 0.631$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a) \pm \epsilon} = n^{0.631 \pm \epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se $f(n) = O(n^{\log_b(a) - \epsilon})$**
    - $f(n) = \log n = O(n^{0.631-\epsilon})$
    - **Como $\epsilon > 0$ gera uma desigualdade verdadeira, o CASO 1 é verdadeiro**
    - **Com isso, o CASO 1 é verdadeiro, o que implica em:** $T(n) = \Theta(n^{\log_b(a)})$
      - $T(n) = \Theta(n^{\log_3(2)})$
- **Com isso, conclui-se que, pelo CASO 1 do Teorema Mestre:**
  - $T(n) = \Theta(n^{\log_3(2)})$

3.12. $T(n) = 2T(\frac{n}{3}) + n$ || $T(n) = \Theta(n)$

- **Primeiro definimos os 3 termos principais**

- $a = 2; b = 3; f(n) = n$
- **Depois calculamos $\log_b(a)$**
  - $\log_3(2) \approx 0.631$
- **Então o substituímos na equação do Teorema Mestre**
  - $n^{\log_b(a) \pm \epsilon} = n^{0.631 \pm \epsilon}$
- **Agora testaremos cada caso**
  - **CASO 1: verificar se $f(n) = O(n^{\log_b(a) - \epsilon})$**
    - $f(n) = n = O(n^{0.631 - \epsilon})$
    - **Como a desigualdade não é válida para todo $\epsilon > 0$, o CASO 1 é falso**
  - **CASO 2: verificar se $f(n) = \Theta(n^{\log_b(a)})$**
    - $f(n) = n = \Theta(n^{0.631})$
    - $n = n^{0.631}$
    - **Como a igualdade não é verdadeira, o CASO 2 é falso**
  - **CASO 3: verificar se $f(n) = \Omega(n^{\log_b(a) + \epsilon})$ e $af(\frac{n}{b}) \leq cf(n)$**
    - **Primeiro, verifica-se $f(n) = \Omega(n^{\log_b(a) + \epsilon})$**
      - $f(n) = n = \Omega(n^{0.631 + \epsilon})$
      - **Agora, deve-se buscar um $\epsilon > 0$ que satisfaça a equação**
        - $n^1 \geq n^{0.631 + \epsilon}$
        - $1 \geq 0.631 + \epsilon$
        - $\epsilon \leq 1 - 0.631$
        - $\epsilon \leq 0.369$
        - Sendo assim, $\epsilon \in]0, 0.369]$
        - Escolhendo um valor para $\epsilon = 0.2$
        - $n = \Omega(n^{0.631 + 0.2})$
        - $n = \Omega(n^{0.831})$
        - **Como a igualdade é verdadeira, a primeira parte do CASO 3 é verdadeira**
    - **Agora, verifica-se $af(\frac{n}{b}) \leq cf(n)$**
      - $2 * f(\frac{n}{3}) \leq c * f(n)$
      - $2 * \frac{n}{3} \leq c * n$
      - $\frac{2n}{3} \leq c * n$
      - **Se considerarmos $c = \frac{2}{3}$, temos que:**
        - $\frac{2n}{3} \leq \frac{2}{3} * n$
        - **Com isso, o CASO 3 é verdadeiro, o que implica em: $T(n) = \Theta(f(n))$**
          - $T(n) = \Theta(n)$
- **Com isso, conclui-se que, pelo CASO 3 do Teorema Mestre:**
  - $T(n) = \Theta(n)$

Exercício 4. Determine um limite assintótico para $T(n) = 2T(\sqrt{n})$. Dica: Faça uma substituição de variável. Faça $m = \log n$

Resolvendo com alteração de variável

- Aplicando a dica:
  - $m = \log n \Leftrightarrow 2^m = n$
- Usando a definição de raiz quadrada:
  - $\sqrt{n} = n^{\frac{1}{2}}$
- Unindo ambos:
  - $\sqrt{n} = (n)^{\frac{1}{2}} = (2^m)^{\frac{1}{2}} = 2^{\frac{m}{2}}$

$Eq.1 : T(n) = 2T(\sqrt{n})$

Substituindo $\sqrt{n}$ por $n^{\frac{1}{2}}$ na $Eq.1$:

$Eq.2 : T(n) = 2T(n^{\frac{1}{2}})$

Então, se tivermos na equação $T(n) = 2T(\sqrt{n})$ um valor de n tal que $n = 2^m$, então teremos que:

- $Eq.3 : T(2^m) = 2T(2^{\frac{m}{2}})$

Criemos arbitrariamente uma nova função de conversão $Eq.4 : T(n) = T(2^m) = R(m)$.

Para que consigamos obter um $T(2^{\frac{m}{2}})$ encontrado na $Eq.2$, precisaríamos que na função R, o parâmetro fosse $\frac{m}{2}$, então:

- $Eq.5 : R(\frac{m}{2}) = T(2^{\frac{m}{2}})$

Assim, substituindo os termos da $Eq.3$ pelos da $Eq.4$ e $Eq.5$, temos que:

- $R(m) = 2R(\frac{m}{2})$

Assim, podemos agora aplicar o método de preferência para a resolução de recorrências.

Teorema Mestre

- $a = 2; b = 2; f(n) = 0$
- $\log_b(a) = \log_2(2) = 1$
- Caso 1
    - $f(n) = O(n^{\log_b(a)-\epsilon})$
    - $0 = O(n^{1-\epsilon})$
    - Para $\epsilon = 1$, temos que:
    - $0 = O(n^{1-1})$
    - $0 = O(n^0)$
    - $0 = O(1)$

Como o caso 1 é verdadeiro, temos que:

- $R(m) = \Theta(m^{\log_b(a)})$
- $R(m) = \Theta(m^1)$
- $R(m) = \Theta(m)$

Substituindo m pelo seu valor inicial $(\log n)$, e utilizando a definição da $Eq.4$ temos que:

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

Exercício 5. Determine um limite assintótico para $T(n) = 2T(\sqrt{n}) + \log n$. Dica: Faça uma substituição de variável. Faça $m = \log n$

Resolvendo com alteração de variável Q5

- Aplicando a dica:
    - $m = \log n \Leftrightarrow 2^m = n$
- Usando a definição de raiz quadrada:
    - $\sqrt{n} = n^{\frac{1}{2}}$
- Unindo ambos:
    - $\sqrt{n} = (n)^{\frac{1}{2}} = (2^m)^{\frac{1}{2}} = 2^{\frac{m}{2}}$

$Eq.1 : T(n) = 2T(\sqrt{n}) + \log n$

Substituindo $\sqrt{n}$ por $n^{\frac{1}{2}}$ na $Eq.1$:

$Eq.2 : T(n) = 2T(n^{\frac{1}{2}}) + \log n$

Então, se tivermos na equação $T(n) = 2T(\sqrt{n}) + \log n$ um valor de n tal que $n = 2^m$, então teremos que:

- $Eq.3 : T(2^m) = 2T(2^{\frac{m}{2}}) + \log(2^m)$
- $\log(2^m) \Leftrightarrow 2^x = 2^m \implies x = m$
- $\log(2^m) = m$
- $Eq.4 : T(2^m) = 2T(2^{\frac{m}{2}}) + m$

Criemos arbitrariamente uma nova função de conversão $Eq.5 : T(n) = T(2^m) = R(m)$.

Para que consigamos obter um $T(2^{\frac{m}{2}})$ encontrado na $Eq.3$, precisaríamos que na função R, o parâmetro fosse $\frac{m}{2}$, então:

- $Eq.6 : R(\frac{m}{2}) = T(2^{\frac{m}{2}})$

Assim, substituindo os termos da $Eq.4$ pelos da $Eq.5$ e $Eq.6$, temos que:

- $Eq.7 : R(m) = 2R(\frac{m}{2}) + m$

Assim, podemos agora aplicar o método de preferência para a resolução de recorrências.

Teorema Mestre Q5

- Teorema Mestre
    - Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função, e $T(n) = aT(\frac{n}{b}) + f(n)$, então, para algum $\epsilon > 0$:
        - Se $f(n) = O(n^{\log_b(a)-\epsilon}) \implies T(n) = \Theta(n^{\log_b(a)})$ [<=]
        - Se $f(n) = \Theta(n^{\log_b(a)}) \implies T(n) = \Theta(n^{\log_b(a)} * log(n))$ [=]
        - Se $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ e $af(\frac{n}{b}) \leq cf(n)$ então $\implies T(n) = \Theta(f(n))$ [>=]

$$\log_a b = x \Leftrightarrow a^x = b$$

- $a =$ Base
- $b =$ Logaritmando
- $x =$ Logaritmo

- $a = 2; b = 2; f(m) = m$
- $\log_b(a) = \log_2(2) = 1$
- Caso 1
  - $f(m) = O(m^{\log_b(a)-\epsilon})$
  - $m = O(m^{1-\epsilon})$
  - $m^1 = O(m^{1-\epsilon})$
    - Caso 1 inválido visto que o menor dos ϵ ainda assim não será válido.
- Caso 2
  - $f(m) = \Theta(m^{\log_b(a)})$
  - $m = \Theta(m^1)$
  - $m = \Theta(m)$
  - 👍

Como o Caso 2 é verdadeiro, temos que:

- $R(m) = \Theta(m^{\log_b(a)} \cdot \log m)$
- $R(m) = \Theta(m^1 \cdot \log m)$
- $R(m) = \Theta(m \cdot \log m)$
- $R(m) = \Theta(m \log m)$

Substituindo m pelo seu valor inicial ($\log n$), e utilizando a definição da $Eq.5$ temos que:

- $T(n) = \Theta((\log n) \log(\log n))$
- $T(n) = \Theta(\log n \log^2 n)$

Exercício 6. Determine e prove uma equivalência assintótica para $T(n) = T(\frac{n}{4}) + T(\frac{n}{5}) + T(\frac{n}{6}) + n$

- **Dica 1:** Divida a prova em limite inferior e limite superior;
- **Dica 2:** Aproxime a função para baixo e para cima usando seus próprios termos.

- Teorema Mestre
  - Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função, e $T(n) = aT(\frac{n}{b}) + f(n)$, então, para algum $\epsilon > 0$:
    - Se $f(n) = O(n^{\log_b(a)-\epsilon}) \implies T(n) = \Theta(n^{\log_b(a)})$ [<=]
    - Se $f(n) = \Theta(n^{\log_b(a)}) \implies T(n) = \Theta(n^{\log_b(a)} * log(n))$ [=]
    - Se $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ e $af(\frac{n}{b}) \leq cf(n)$ então $\implies T(n) = \Theta(f(n))$ [>=]

- $T(n) = T(\frac{n}{4}) + T(\frac{n}{5}) + T(\frac{n}{6}) + n$
- $T^+(n) = T(\frac{n}{4}) + T(\frac{n}{4}) + T(\frac{n}{4}) + n$
  - Consideraremos que $T^+(n)$ como sendo uma função similar que seja **maior** que $T(n)$
  - $T^+(n) = 3T(\frac{n}{4}) + n$
- $T^-(n) = T(\frac{n}{6}) + T(\frac{n}{6}) + T(\frac{n}{6}) + n$
  - Consideraremos que $T^-(n)$ como sendo uma função similar que seja **menor** que $T(n)$
  - $T^-(n) = 3T(\frac{n}{6}) + n$
- Executando o Teorema Mestre para $T^+(n)$:
  - $a = 3; b = 4; f(n) = n$
  - $\log_b(a) = \log_4(3) = 0,792$
  - Caso 1:
    - $n^{\log_b(a)-\epsilon} = n^{\log_4(3)-\epsilon}$
    - $f(n) = n = O(n^{\log_4(3)-\epsilon})$
    - $n^1 = O(n^{\log_4(3)-\epsilon})$
    - $1 < \log_4(3) - \epsilon$
    - $\epsilon < 0,792 - 1$

- $\epsilon < -0,208$
- $0 < \epsilon < -0,208$ 👎
- Com isso, entendemos que o caso 1 não é válido

- Caso 2:
  - $n^{\log_b(a)} = n^{\log_4(3)}$
  - $f(n) = n = \Theta(n^{\log_4(3)})$
  - $n^1 = \Theta(n^{0,792})$
- Executando o Teorema Mestre para $T^-(n)$:
  - $a = 3; b = 6; f(n) = n$
  - $\log_b(a) = \log_6(3)$
  - Caso 1:
    - $n^{\log_b(a)-\epsilon} = n^{\log_6(3)-\epsilon}$
    - $f(n) = n = O(n^{\log_6(3)-\epsilon})$
    - $n^1 = O(n^{\log_6(3)-\epsilon})$ [#confia]
    - $1 = \log_6(3) - \epsilon$
    - $\epsilon = \log_6(3) - 1$
    - Com isso:
      - $T^-(n) = \Theta(n^{\log_6(3)})$

Então, considerando a equivalência assintótica para $T^+(n)$ e $T^-(n)$, desconsideraremos a base do $\log$, enfim tendo que:

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

Complexidade Amortizada

Para as questões a seguir considere uma pilha S que possui duas operações

- **pop(S)**: remove (desempilha) o topo da pilha S.
- **push(S,x)**: empilha o elemento x na pilha S.

Cada uma dessas operações possui custo O(1). Vamos definir uma nova operação para esta estrutura, a operação **multi-pop(S,k)** que remove os k últimos elementos empilhados.

Exercício 7. Apresente um algoritmo para a operação de multi-pop. Observe que você pode usar a operação inicial de pop e a operação vazio(S) (que verifica se a pilha S é vazia ou não) no seu algoritmo

---

- Algoritmo para **multi-pop**:
  - Entrada: pilha S e inteiro k
  - $restaRemover \leftarrow k$
  - **ENQUANTO** !vazio(S) **E** restaRemover > 0 **FAÇA:**
    - pop(S)
    - $restaRemover \leftarrow restaRemover - 1$

Exercício 8. Qual a complexidade amortizada da operação de multi-pop dada uma sequência de operações de push, pop e multi-pop em uma pilha originalmente vazia?

Complexidade multi-pop:

- Algoritmo para **multi-pop**:
  - Entrada: pilha S e inteiro k
  - $restaRemover \leftarrow k$ $[C_1]$
  - **ENQUANTO** !vazio(S) **E** restaRemover > 0 **FAÇA:** $[C_2]$
    - pop(S) $[C_3]$
    - $restaRemover \leftarrow restaRemover - 1$ $[C_4]$
- $C_1 : 1 = O(1)$
- $C_2 : O(1) + 1 = O(1)$ [Considerando que O(vazio(S)) = O(1)]
- $C_3 : 1 = O(1)$
- $C_4 : 2 = O(1)$

Através do método contável, consideraremos que no pior caso, teremos n operações de **push**. Então, como cada operação de **push** tem complexidade $O(1)$, a complexidade total será de $O(n)$.

Então, ao fazermos a complexidade do pior caso, divida pelo número de operações, teremos a complexidade amortizada.

- Análise de complexidade amortizada: $\frac{O(n)}{n} = O(1)$

Exercício 9. Qual o custo computacional de sequência de n operações de push, pop e multi-pop em uma pilha com inicialmente s_O elementos e que termina com s_n elementos?

Considerarei que serão realizados separadamente n operações de push; n operações de pop; e n operações de multi-pop.

- **push**: adiciona um elemento
- **pop**: remove um elemento
- **multi-pop**: remove k elementos

Determinemos que:

- S_O é a quantidade inicial de elementos na pilha
- S_n é a quantidade final de elementos na pilha
- Pu é a quantidade de operações de push
- Po é a quantidade de operações de pop
- M é a quantidade de operações de multi-pop
- n é a quantidade total de operações realizadas, sendo ela igual à soma de operações de push, pop e multi-pop.
  - Ou seja, $n = P + O + M$

Então após n operações distribuídas aleatoriamente entre Pu operações de Push, Po operações de Pop e M operações de Multi-Pop, teremos que $n = Pu + Po + M$. A complexidade de cada operação é $O(1)$, então a complexidade de Pu operações de Push é $Pu * O(1) = O(Pu)$, a complexidade de Po operações de Pop é $Po * O(1) = O(Po)$ e a complexidade de M operações de Multi-Pop é $M * O(1) = O(M)$.

Então, considerando também que S_n é a quantidade de elementos após as n operações, Podemos dizer que $S_n = S_O + Pu - Po - M$. Então a variação de elementos será dado por $S_n - S_O = Pu - Po - M$.

Sendo assim, o custo computacional será de $O(P) + O(O) + O(M) = O(P + O + M) = O(n)$.