Lista 2 - MAC338

João Gabriel Basi - N° USP: 9793801

1. **a)** Supondo que n é potência de 2 e adotando n^2 como representante da classe $\Theta(n^2)$, temos:

$$\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + n^2 \\
&= 2\left(2T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}\right) + n^2 = 2^2T\left(\frac{n}{2^2}\right) + n^2\left(1+\frac{1}{2}\right) \\
&= 2^2\left(2T\left(\frac{n}{2^3}\right) + \frac{n^2}{2^4}\right) + n^2\left(1+\frac{1}{2}\right) = 2^3T\left(\frac{n}{2^3}\right) + n^2\left(1+\frac{1}{2}+\frac{1}{2^2}\right) \\
&= ... \\
&= 2^kT\left(\frac{n}{2^k}\right) + n^2\sum_{i=0}^{k-1}\frac{1}{2^i} \overset{PG}{=} 2^kT\left(\frac{n}{2^k}\right) + 2n^2\left(1-\frac{1}{2^k}\right)
\end{aligned}$$

Supondo $k = \lg n$ e $T(1) = 1$, obtemos:

$$\begin{aligned}
T(n) &= nT(1) + 2n^2\left(1-\frac{1}{n}\right) \\
&= n + 2n^2 - 2n \\
&= 2n^2 - n \\
&= \Theta(n^2)
\end{aligned}$$

Verifcando o resultado:
Reescrevendo-o em termos de $k = \lg n$ temos

$$\begin{aligned}
T(2^k) &= 2\cdot(2^k)^2 - 2^k \\
&= 2^{2k+1} - 2^k
\end{aligned}$$

Para $k = 0$: $T(2^0) = 2^{2\cdot 0+1} - 2^0 = 2 - 1 = 1$
Para $k \geqslant 1$, supomos que a fórmula vale para $k-1$. Então, utilizando a recorrência, temos:

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

Portanto concluímos que $T(n) = \Theta(n^2)$.

d) Supondo que n é potência de 3 e adotando n^2 como representante da classe $\Theta(n^2)$, temos:

$$
\begin{aligned}
T(n) &= 7T\left(\frac{n}{3}\right) + n^2 \\
&= 7T\left(7T\left(\frac{n}{3^2}\right) + \frac{n^2}{3^2}\right) + n^2 = 7^2T\left(\frac{n}{3^2}\right) + n^2\left(1 + \frac{7}{9}\right) \\
&= 7^2T\left(7T\left(\frac{n}{3^3}\right) + \frac{n^2}{3^4}\right) + n^2\left(1 + \frac{7}{9}\right) = 7^3T\left(\frac{n}{3^3}\right) + n^2\left(1 + \frac{7}{9} + \left(\frac{7}{9}\right)^2\right) \\
&= \ldots \\
&= 7^kT\left(\frac{n}{3^k}\right) + n^2\sum_{i=0}^{k-1}\left(\frac{7}{9}\right)^k \overset{PG}{=} 7^kT\left(\frac{n}{3^k}\right) + \frac{9n^2\left(1 - \left(\frac{7}{9}\right)^k\right)}{2}
\end{aligned}
$$

Supondo $k = \log_3 n$ e $T(1) = 1$ e sabendo que $a^{\log_b c} = c^{\log_b a}$, obtemos:

$$
\begin{aligned}
T(n) &= 7^{\log_3 n}T(1) + \frac{9n^2\left(1 - \left(\frac{7}{9}\right)^{\log_3 n}\right)}{2} \\
&= n^{\log_3 7} + \frac{9n^2(1 - n^{\log_3 7 - 2})}{2} \\
&= \frac{2n^{\log_3 7} + 9n^2 - 9n^{\log_3 7}}{2} \\
&= \frac{9n^2 - 7n^{\log_3 7}}{2} \\
&= \Theta(n^2)
\end{aligned}
$$

Verifcando o resultado:
Reescrevendo-o em termos de $k = \log_3 n$ temos

$$
\begin{aligned}
T(3^k) &= \frac{9 \cdot 3^{2k} - 7 \cdot 3^{k \log_3 7}}{2} \\
&= \frac{3^{2k+2} - 7^{k+1}}{2}
\end{aligned}
$$

Para $k = 0$: $T(3^0) = \frac{3^{2 \cdot 0+2} - 7^{0+1}}{2} = \frac{9-7}{2} = 1$
Para $k \geqslant 1$, supomos que a fórmula vale para $k - 1$. Então, utilizando a recorrência, temos:

$$
\begin{aligned}
T(3^k) &= 7T(3^{k-1}) + 3^{2k} \\
&= 7\left(\frac{3^{2k} - 7^k}{2}\right) + 3^{2k} \\
&= \frac{7 \cdot 3^{2k} - 7^{k+1} + 2 \cdot 3^{2k}}{2} \\
&= \frac{3^{2k+2} - 7^{k+1}}{2}
\end{aligned}
$$

Portanto concluímos que $T(n) = \Theta(n^2)$.

2. Pseudo-código do merge:

```
TMERGESORT(A, b, e):
   se e-b > 2, então:
      p ← (e-b)/3
      m1 ← b + ⌈p⌉
      m2 ← b + ⌈2*p⌉
      TMERGESORT(A, b, m1)
      TMERGESORT(A, m1, m2)
      TMERGESORT(A, m2, e)
      INTERCALA(A, b, m1, m2)
      INTERCALA(A, b, m2, e)
   senão, se e-b == 2 e A[b] > A[b+1], então:
      A[b] ↔ A[b+1]
```

Pseudo-código do intercala:

```
INTERCALA(A, b, m, e):
   B[0...e-b]
   para i ← 0 até m-b-1 faça:
      B[i] ← A[b+i]
   para i ← m-b até e-b-1 faça:
      B[i] ← A[e+m-b-i-1]
   i ← 0
   j ← e-b-1
   para k ← b até e-1 faça:
      se B[i] ⩽ B[j], então:
         A[k] ← B[i]
         i ← i + 1
      senão
         A[k] ← B[j]
         j ← j - 1
```

Análise linha por linha:

INTERCALA

$$\begin{array}{lcl} 3-6 & = & \Theta(n) \\ 7-8 & = & \Theta(1) \\ 9-15 & = & \Theta(n) \\ \hline T(n) & = & \Theta(n) \end{array}$$

TMERGESORT

$$\begin{array}{lcl} 2-5 & = & \Theta(1) \\ 6 & = & T\left(\left\lceil \frac{n}{3} \right\rceil\right) \\ 7 & = & T\left(\left\lceil \frac{n}{3} \right\rceil\right) \\ 8 & = & T\left(\left\lfloor \frac{n}{3} \right\rfloor\right) \\ 9-10 & = & \Theta(n) \\ 11-12 & = & \Theta(1) \\ \hline T(n) & = & \Theta(n) + 2T\left(\left\lceil \frac{n}{3} \right\rceil\right) + T\left(\left\lfloor \frac{n}{3} \right\rfloor\right) \end{array}$$

A partir das contas acima, chegamos na recorrência

$$T(n) = \Theta(n) + 2T(\left\lceil \frac{n}{3} \right\rceil) + T(\left\lfloor \frac{n}{3} \right\rfloor)$$

Resolvendo-a para n potência de 3 e $k = log_3 n$ temos

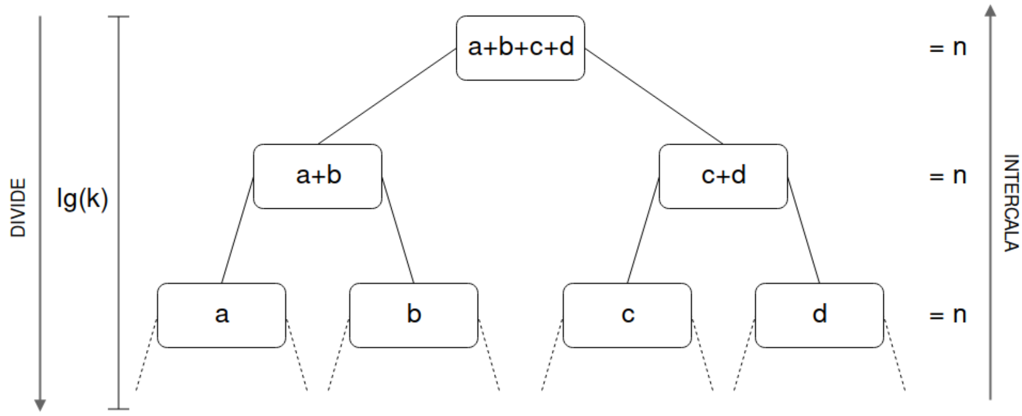
$$T(n) = n + n\log_3 n = \Theta(n\log_3 n)$$
$$T(3^k) = 3^k + k3^k$$

Verifcando o resultado:
Para $k = 0$: $T(3^0) = 3^0 + 0 \cdot 3^0 = 1 + 0 = 1$
Para $k \geqslant 1$, supomos que a fórmula vale para $k-1$. Então, utilizando a recorrência, temos:

$$\begin{aligned} T(3^k) &= 3T(3^{k-1}) + 3^k \\ &= 3(3^{k-1} + (k-1)3^{k-1}) + 3^k \\ &= 3^k + (k-1)3^k + 3^k \\ &= 3^k + k3^k \end{aligned}$$

Portando concluímos que o algoritmo é $\Theta(n\log_3 n)$.

4. O algoritmo terá duas funções, a DIVIDE e a INTERCALA. A DIVIDE recebrá uma lista de listas, dividirá a lista em duas partes com quase o mesmo tamanho, executará DIVIDE em cada parte e devolverá o resultado da INTERCALA entre as duas listas devolvidas pelas execuções do DIVIDE. A INTERCALA receberá duas listas ordenadas e devolverá uma lista ordenada com todos os elementos das listas recebidas.
Representando como a, b, c e d os tamanhos das listas recebidas, temos a seguinte árvore de recorrência:



A cada chamada do DIVIDE, as listas são divididas em dois grupos, o que faz com que a árvore tenha tamanho médio $\lg k$ e, a cada nível da árvore, o tempo gasto pelo INTERCALA a cada execução (que está anotado nos quadrinhos da figura) soma n, pois sabemos que $a + b + c + d = n$ e que a INTERCALA é linear. Com isso, podemos concluir que o algoritmo é $\Theta(n \lg k)$. Em especial se $k = 2$, a complexidade é $n \lg 2 = n = \Theta(n)$ e se $k = n$ a complexidade é $n \lg n = \Theta(n \lg n)$.