Resolução 2ª Lista de Exercícios 2022

Questão 1

	int Peso (int *V, int ini, int fim)	
	{	
1	`int i, P=0;`	1
2	`for(i=ini;i<=fim;i++)`	2n + 2
3	`P += V[i];`	2n
4	`return P;`	1
	}	

i = i + 1

P = P + V [i] → 3

j++;

$\underbrace{4n}+4$

j = j + 1

```
int Falsa (int *V, int ini, int fim)
{
        if (ini == fim)
                return ini;
        else
        {
                int meio = (ini+fim)/2;
                if (Peso(V,ini,meio) < Peso(V,meio+1,fim))
                        return Falsa(V,ini,meio);
                else
                        return Falsa(V,meio+1,fim);
        }
}
```

n

1

1

3

$4 + 2\left(\frac{4n}{2} + 4\right)$

3

Recursão

$1 + 3 + 4 + 4n + 8 + 3$

$4n + 19$

Caso base $\begin{cases} T(n) = \underline{2} & n = 1 \\ T(n) = T(n/2) + \underbrace{4n + 19} \end{cases}$ Caso Rec.

$O(n)$

$k = 1 \quad T(n) = T(n/2) + 4n + 19$

$T(n) = T(n/4) + 4\frac{n}{2} + 19 + 4n + 19$

$k = 2 \quad T(n) = T(n/4) + 4n + 4\frac{n}{2} + 38$

$T(n) = T(n/8) + 4\frac{n}{4} + 19 + 4n + 4\frac{n}{2} + 38$

$k = 3 \quad T(n) = T(n/8) + 4\frac{n}{1} + 4\frac{n}{2} + 4\frac{n}{4} + 57$

$$T(n) = T\left(\frac{n}{2^{\log_2 n}}\right) + \sum_{i=0}^{\log_2 n - 1} \frac{4n}{2^i} + \lg \log_2 n \qquad k = \log_2 n$$

$$\frac{4n}{1} + \frac{4n}{2} + \frac{4n}{4}$$

$$S_{TPG} = \frac{a_1(1-q^m)}{1-q} \qquad q = 0{,}5$$

$$S_{TPG} = 2\,\frac{(1-0{,}5^{\log_2 n})}{\cancel{0{,}5}}$$

$$S_{TPG} = 2(1-n^{\log_2 0{,}5})$$

$$S_{TPG} = 2(1-n^{-1})$$

$$S_{TPG} = 2\left(1-\frac{1}{n}\right)$$

$$S_{TPG} = 2 - \frac{2}{n}$$

$$4n \sum_{i=0}^{\log_2 n - 1} \frac{1}{2^i} = 4n\left(2 - 2/n\right) = 8n - 8$$

$$T(n) = 2 + 8n - 8 + 19 \log_2 n$$

$$T(n) = 8n + 19 \log_2 n - 6$$

$$O(n) \,//$$

Questão 2

$$T(n) = 1$$
$$T(n) = 3T\left(\frac{n}{3}\right) + 12n + 3$$

$n^{\log_b a} \rightarrow n^{\log_3 3} \rightarrow n^1$

$12n + 3 \rightarrow n$

$> O(n \log_2 n)$

Caso 2

$$T(n) = 1$$
$$T(n) = 4T\left(\frac{n}{2}\right) + 4n$$

$n^{\log_b a} \rightarrow n^{\log_2 4} \rightarrow n^2$

$4n \rightarrow n^1$

Caso 1

$O(n^2)$

$$T(n) = 1$$
$$T(n) = 3T\left(\frac{n}{4}\right) + n$$

$n^{\log_b a} \rightarrow n^{\log_4 3} \rightarrow n^{0,7925}$

$n \rightarrow n$

Caso 3

$O(n)$

$$T(n) = 1$$
$$T(n) = 2T\left(\frac{n}{2}\right) + 7n + 2$$

$n^{\log_b a} \to n^{\log_2 2} \to n$

$7n + 2 \to n$

Caso 2 $\quad O(n \log_2 n)$

$$T(n) = 1$$
$$T(n) = 2T\left(\frac{n}{2}\right) + 3n^2 - 12n + 2$$

$n^{\log_b a} \to n^{\log_2 2} \to n^1$

$3n^2 - 12n + 2 \to n^2$

Caso 3 $\quad O(n^2)$

Questão 3

Algoritmo 1: Divide o problema em 3 partes de tamanho n/4 cada e gasta um tempo adicional de O(1) por chamada.

Algoritmo 2: Divide o problema em 3 partes de tamanho n/2 cada e gasta um tempo adicional de $O(n^2)$ por chamada.

Algoritmo 3: Divide o problema em 3 partes de tamanho n/3 cada e gasta um tempo adicional de O(n) por chamada.

Resp:

Letra A

$A_1 \quad 3T\left(\frac{n}{4}\right) + 1 \rightarrow n^{\log_4 3} \rightarrow n^{0,79\;5}$

$A_2 \quad 3T(n/2) + n^2 \rightarrow n^2 > n^{\log_2 3} \rightarrow n^2$

$A_3 \quad 3T(n/3) + n \rightarrow n = n \rightarrow n \log n$

Questão 4

a) $$T(n) = 1$$
$$T(n) = T(n-1) + 4$$

$K=1$ $\quad T(n) = T(n-1) + 4$

$T(n) = T(n-1-1) + 4 + 4$

$K=2$ $\quad T(n) = T(n-2) + 8$

$T(n) = T(n-2-1) + 4 + 8$

$K=3$ $\quad T(n) = T(n-3) + 12$

$T(n) = T(n-K) + 4K$

$n - K = 1$

$K = n - 1$

$T(n) = T(n-(n-1)) + 4(n-1)$

$T(n) = T(1) + 4n - 4$

$T(n) = 1 + 4n - 4$

$T(n) = 4n - 3$

$O(n)$

b) $T(n) = 1$
$T(n) = T(n-1) + n$

$k=1$ $\quad T(n) = T(n-1) + n \qquad T(n-1) = T(n-2) + n - 1$

$T(n) = T(n-2) + n - 1 + n$

$k=2$ $\quad T(n) = T(n-2) + 2n - 1 \qquad T(n-2) = T(n-3) + n - 2$

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

$k=3$ $\quad T(n) = T(n-3) + 3n - 3 \qquad T(n-3) = T(n-4)\ n - 3$

$T(n) = T(n-4) + n - 3 + 3n - 3$

$k=4$ $\quad T(n) = T(n-4) + 4n - 6$

$$T(n) = T(n-k) + kn + \sum_{i=0}^{k-1} -i$$

$n - k = 1$

$k = n - 1$

$$T(n) = T(n-n+1) + n(n-1) + \sum_{i=0}^{n-2}$$

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

$T(n) = n^2 - 2n + 2$

$O(n^2)$

c) $T(n) = 1$

$T(n) = 2T\left(\frac{n}{2}\right) + 7n + 2$

$T(n/2) = 2T(n/4) + 7\frac{n}{2} + 2$

$k=1 \quad T(n) = 2T\left(\frac{n}{2}\right) + 7n + 2$

$T(n) = 2\left(2T\left(\frac{n}{4}\right) + 7\frac{n}{2} + 2\right) + 7n + 2$

$T(n/4) = 2T(n/8) + 7\frac{n}{4} + 2$

$k=2 \quad T(n) = 4T\left(\frac{n}{4}\right) + 14n + 6$

$n/2^k = 1$

$T(n) = 4\left(2T\left(\frac{n}{8}\right) + \frac{7n}{4} + 2\right) + 6 + 14n$

$n = 2^k$

$k = \log_2 n$

$k=3 \quad T(n) = 8T\left(\frac{n}{8}\right) + 21n + 14$

$T(n) = 2^k T\left(\frac{n}{2^k}\right) + 7kn + 2 \cdot 2^k - 2$

$$T(n) = 2^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + 7n\log_2 n + 2 \cdot 2^{\log_2 n} - 2$$

$$T(n) = n^{\log_2 2} T\left(\frac{n}{n^{\log_2 2}}\right) + 7n\log_2 n + 2n^{\log_2 2} - 2$$

$$T(n) = n \cdot T(1) + 7n\log_2 n + 2n - 2$$

$$T(n) = 7n\log_2 n + 3n - 2$$

$$O(n\log_2 n)$$

Questão 5

T(n) = 8T(n/2)+q*n se n > 1. Dado que T(1) = p, e que p e q são constantes arbitrárias

Teorema mestre

$n^{\log_b a} \rightarrow n^{\log_2 8} \rightarrow n^3$

$q * n \rightarrow n$

Caso 1:

$O(n^3)$

Resp. Letra D

Questão 6

A cada iteração pega-se um Menor(n) e Maior(n), cada um com custo n-2. Assim o custo $f(n) = 2(n-1)$, $f(n) = 2n - 2$. A cada iteração temos um novo problema com 2 unidades a menos, assim:

$$T(n) = T(n-2) + 2n - 2$$

Resp. Letra B

Questão 7

$$T(n) = T(n-2) + 2n - 2$$

Cada iteração tem 2 elementos a menos. Como maior e menor têm custos n-1 temos 2 custos n-1, logo $2(n-1) = 2n-2$

$k=1$ $\quad T(n) = T(n-2) + 2n - 2$ $\qquad T(n-2) = T(n-2-2) + 2(n-2) - 2$

$\qquad T(n) = T(n-4) + 2n + 2n - 6 - 2$ $\qquad T(n-2) = T(n-4) + 2n - 6$

$k=2$ $\quad T(n) = T(n-4) + 4n - 8$ $\qquad T(n-4) = T(n-4-2) + 2(n-4) - 2$

$\qquad T(n) = T(n-6) + 2n - 10 - 4n - 8$ $\qquad T(n-4) = T(n-6) + 2n - 10$

$k=3$ $\quad T(n) = T(n-6) + 6n - 18$

Analisando os termos frente a k temos

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

$k=3$

$4 \cdot 0 + 2 = 2$
$4 \cdot 1 + 2 = 6$
$4 \cdot 2 + 2 = 10$

18

Considerando que n é par ele vai entrar no segundo caso base $j-i=1$, Digamos que $n=16$. Quando $n=2$ ele já pára. Nesse caso temos:

$$n - 2k = 2$$

$$n = 2 + 2k$$

$$\frac{n-2}{2} = k$$

$$k = n/2 - 1$$

Considerem que $n = 8$

A cada iteração diminui 2 de n

7	5	2	1	3	4	8	6

Maior = 8
Menor = 1

5	2	7	3	4	6

Maior = 7
Menor = 2

5	3	4	6

Maior = 6
Menor = 3

Casos recursivos

5	4

→ Caso base $n = 2$

Por questão de simplicidade, podemos considerar que n é uma potência de 2, assim sempre vai entrar no caso base $j-i=1$.

Se n for ímpar vai entrar no outro caso.

Eu usaria o primeiro cenário

$$T(n) = T\left(n-2\left(\frac{n}{2}-1\right)\right) + 2n\left(\frac{n}{2}-1\right) - 4\sum_{i=0}^{n/2-1-2} i - 2\left(\frac{n}{2}-1-1\right)$$

$$T(n) = T(n-n+2) + n^2 - 2n - 4\,(\text{Soma Termos PA}) - n + 2$$

$$STPA = \frac{(a_1 + a_n)\,n}{2}$$

$$STPA = \frac{\left(0+\frac{n}{2}-2\right)\left(\frac{n}{2}-1\right)}{2}$$

$$STPA = \frac{n^2}{4} - \frac{n}{2} - n + 2 \Big/ 2$$

$$STPA = \frac{n^2}{8} - \frac{3n}{4} + 1$$

$$T(n) = T(2) + n^2 - 2n - 4\left(\frac{n^2}{8} - \frac{3n}{4} + 1\right)$$

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

$$T(n) = \frac{n^2}{2} + n + c - 4$$

$$O(n^2)$$

Questão 8

```
void Imprime (int *V, int tam, int ini, int fim)
{
```

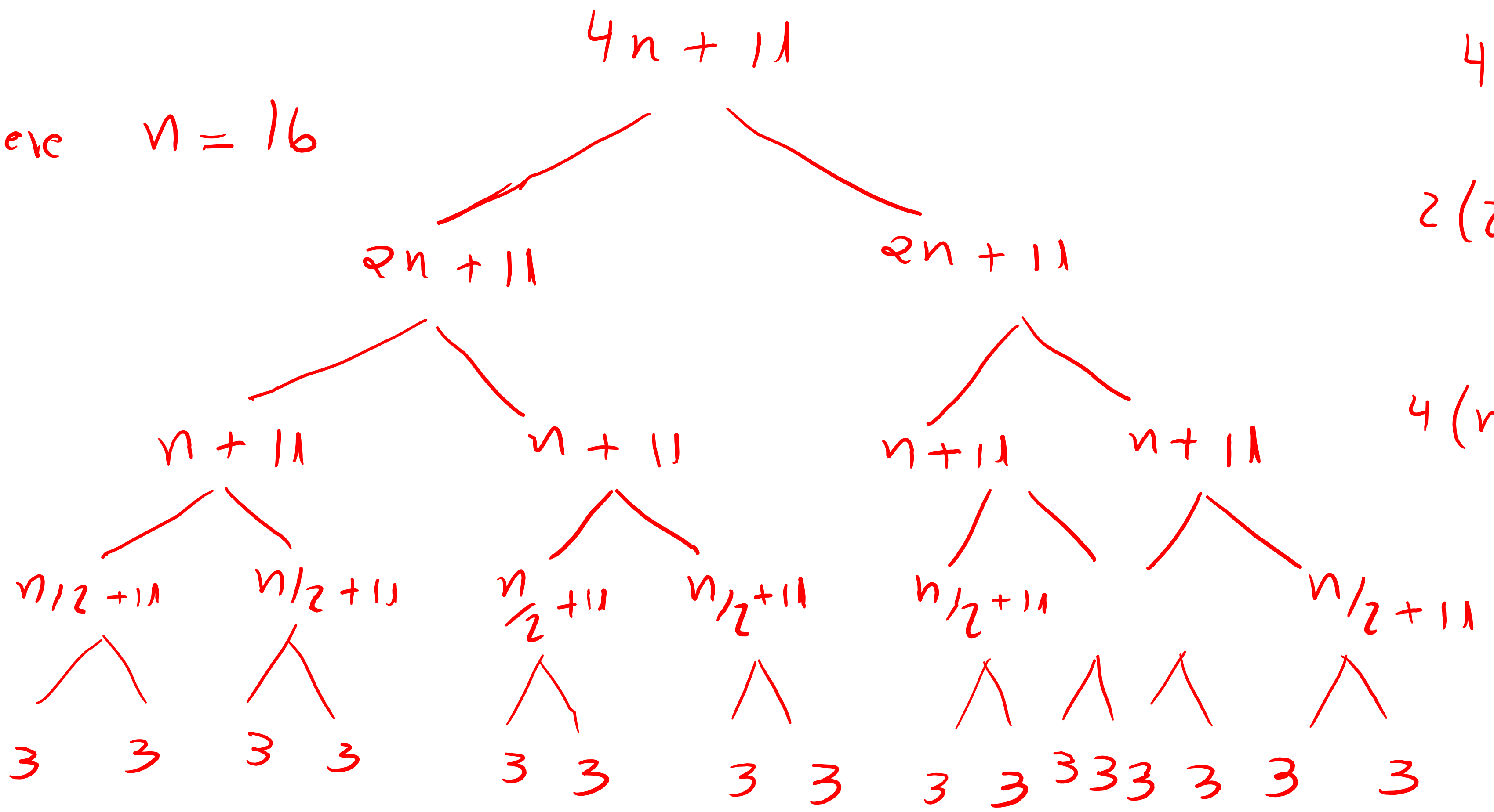
`int i;`	
`if (tam == 1)`	1
`    printf(V[ini]);`	2
`else`	
`{`	
`    for(i=0;i<tam;i++)`	$2n + 2$
`        printf("%d",V[i]);`	$2n$
`    int meio = (ini+fim)/2;`	3
`    Imprime(V,tam/2,ini,meio);`	2
`    Imprime(V,tam/2,meio+1,fim);`	3
`}`	
`}`	

Caso base $\quad T(n) = 3 \quad n = 1$

Caso Recursivo $\quad T(n) = 2T\left(\frac{n}{2}\right) + 4n + 11$

Cada vez que uma chamada recursiva é feita paga-se o custo $f(n)$ proporcional ao tamanho de n

Considere $n = 16$



$4n + 11$

$2(2n + 11) = 4n + 22$

$4(n + 11) = 4n + 44$

$8\left(\frac{n}{2} + 11\right) = 4n + 88$

$16(3) = 48$

$\hookrightarrow 3n$

O custo de cada nível é

$4n + 2^{\text{nível}} \cdot 11$

a altura da árvore é $\log_2 n$ (arv. binária)

$CT = CI + CF$

$$CT = \text{altura} \cdot 4n + \underbrace{\sum_{i=0}^{\text{altura}-1} 2^i \cdot 11}_{} + 3n$$

$CT = 4n \log_2 n + 11(n-1) + 3n$

$CT = 4n \log_2 n + 11n - 11 + 3n$

$CT = 4n \log_2 n + 14n - 11$

$O(n \log_2 n)$