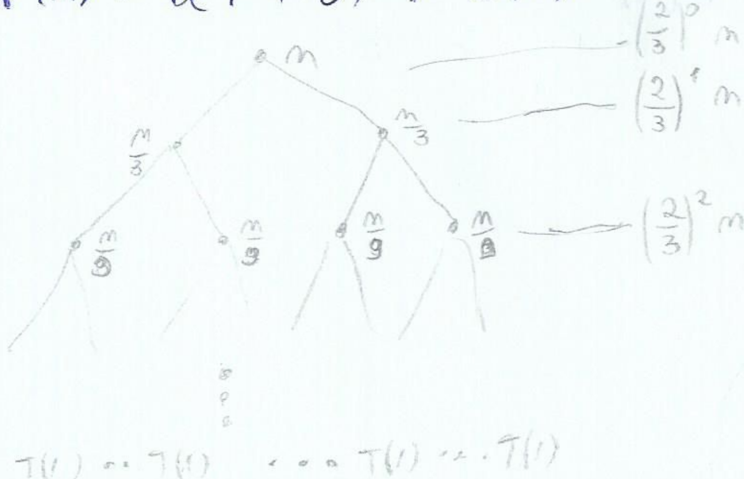
⑥ (e) $T(n) = 2T(n/3) + \Theta(n)$



$$\frac{n}{3^i} = 1$$

$$n = 3^i \Leftrightarrow i = \log_3 n$$

Niveis: $\log_3 n + 1$

Soma Geometrica

$$S = \frac{(1^{o}\ termo).(Razão^n - 1)}{Razão - 1}$$

$$S = \frac{(1).\left(\frac{2}{3}^{\log_3 n + 1} - 1\right)}{\frac{2}{3} - 1}$$

$$S = \frac{\left(\frac{2}{3}^{\log_3 n}\right)\left(\frac{2}{3}\right) - 1}{-\frac{1}{3}}$$

$$S = \left(-\frac{3}{1}\right)\left(\frac{2^{\log_3 n}}{3^{\log_3 n}}\right)\left(\frac{2}{3}\right) + 3$$

$$S = \frac{2^{\log_3 n}}{3^{\log_3 n}} \cdot (-2) + 3$$

$$S = \frac{n^{\log_3 2}}{n^{\log_3 3}} \cdot (-2) + 3$$

$$S = \frac{n^{\log_3 2}}{n} (-2) + 3$$

$$S = \frac{2n^{\log_3 2}}{n} + 3$$

$$T(n) = n + \frac{2}{3}n + \frac{4}{9}n + \dots$$

$$= n\left[\left(\frac{2}{3}\right)^0 + \left(\frac{2}{3}\right)^1 + \left(\frac{2}{3}\right)^2 + \dots\right]$$

$$T(n) = \cancel{n}\left[\frac{2n^{\log_3 2}}{\cancel{n}} + 3\right]$$

$$T(n) = 2n^{\log_3 2} + 3n$$

Portanto

$$T(n) \in O(n^{\log_3 2})$$

⑦ (a) Algoritmo A = Divide em 5 subproblemas de metade do tamanho $5T(\frac{n}{2})$ combina soluções em $O(n)$

$$T(n) = 5T(\tfrac{n}{2}) + O(n)$$

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

$a = 5$ $b = 2$ $\log_b a = \log_2 5$ $f(n) = n$

Caso 1. $f(n) \in O(n^{(\log_b a) - \varepsilon})$ para $\varepsilon > 0$

$$n \in O(n^{(\log_2 5) - \varepsilon})$$

Portanto $T(n) \in \Theta(n^{\log_2 5})$

(b) Algoritmo B = Divide em dois subproblemas de tamanho $(n-1)$ combina soluções em $f(n) = O(1)$

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

Pelo metodo da expansão telescopica
- Tamanho do passo $T(n-1)$
- Expandindo próximos passos

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

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

7) continuação

-substituindo em $T(n)$

$T(n-1)$.

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

$T(n-2)$

$$T(n) = 2^2(2T(n-3)+1) + 2 + 1$$
$$= 2^3 T(n-3) + \underbrace{2^2 + 2^1 + 2^0}$$

→ serie geometrica

$$S = \frac{(1^{\circ}\text{termo})(\text{Razao}^n - 1)}{\text{Razao} - 1}$$

$$S = \frac{1 \cdot 2^{n-1} - 1}{2-1}$$

$$S = 2^{n-1} - 1$$

-Encontrando iesimo passo

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

-Encontrando i

$$T(n-i) = T(1)$$
$$n - i = 1$$
$$i = n-1$$

-substituindo i na formula

$$T(n) = 2^{n-1} T(n-(n-1)) + 2^{n-1} - 1$$
$$= 2^{n-1} \cdot 1 + 2^{n-1} - 1$$
$$= 2^1 \cdot 2^{n-1} - 1$$
$$\boxed{= 2^n - 1}$$

-Provando por indução

-caso base

$$T(1) = 2^n - 1$$
$$1 = 2^1 - 1$$
$$1 = 1 \checkmark$$

-supondo $T(n-1) = 2^{n-1} - 1$

$$- T(n) = 2T(n-1) + 1$$
$$= 2^1 (2^{n-1} - 1) + 1$$
$$= 2^n - 2 + 1$$
$$\boxed{= 2^n - 1}$$

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

(c) Algoritmo C: Divide em 9 subproblemas de $n/3$ combina soluções em $O(n^2)$

$$T(n) = 9T(n/3) + O(n^2)$$

-Teorema mestre $a = 9$ $b = 3$ $f(n) = O(n^2) = n^2$

$$n^{\log_b a} = n^{\log_3 9} = n^2$$

$$f(n) = n^{\log_b a}$$
$$n^2 = n^2$$

Portanto é caso 2.

então,

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

7 continuação

- consumo de tempo de cada algoritmo

  Algoritmo a) $\Theta(n^{\log_2 5})$

  Algoritmo b) $\Theta(2^n)$

  Algoritmo c) $\Theta(n^2 \log_2 n)$

A complexidade do algoritmo da letra a) é mais eficiente assintoticamente no pior caso, pois sua complexidade do tempo é $T(n) = \Theta(n^{\log_2 5})$. fazendo comparação com os outros algoritmos, ele é melhor pois tem um valor fixo em seu expoente.

8 (a) temos $T(n) = 2T(\frac{n}{2}) + c$

- pelo método da iteração
  - expandindo próximos passos

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

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

  - substituindo na formula

$$T(n) = 2T(\frac{n}{2}) + c$$
$$= 2(2T(\frac{n}{4}) + c) + c$$
$$= 2^2(2T(\frac{n}{8}) + c) + 2c + c$$
$$= 2^3 T(\frac{n}{8}) + 2^2 c + 2c + c$$
$$= 2^3 T(\frac{n}{8}) + c[2^2 + 2^1 + 2^0]$$

- iesimo passo

$$= 2^i T(\frac{n}{2^i}) + c[2^{i-1} + 2^{i-2} + 2^0]$$

- encontrando i

$$T(1) = T(\frac{n}{2^i})$$
$$1 = \frac{n}{2^i}$$
$$i = \log_2 n$$

- Na formula

$$= 2^{\log_2 n} T(\frac{n}{\log_2 n}) + c[n-1]$$
$$= n^{\log_2 2} T(\frac{n}{\log_2 2}) + cn - c$$
$$= n + cn - c$$

$$S = \frac{(1^{\circ}\text{termo}).(\text{Razão}^n - 1)}{\text{Razão} - 1}$$
$$S = \frac{(1)(2^{\log_2 n} - 1)}{2-1}$$
$$S = \frac{2^{\log_2 n} - 1}{1}$$
$$\boxed{= n - 1}$$

- Por indução
  - Base

$T(1) = 1 + 2.1 - 2$ para $c = 2$

$1 = 1$

  - indução

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

  - Tendo

$$T(n) = 2T(\frac{n}{2}) + c$$
$$= 2(\frac{n}{2} + c.\frac{n}{2} - c) + c$$
$$= n + cn - 2c + c$$
$$= n + cn - c$$

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

Alem disso:

Dado $n=2$
- Ainda Rodando

imprime 1 vez

Dado $n=4$
- Ainda Rodando
- Ainda Rodando
- Ainda Rodando

imprime 3 vezes

Assim temos $(n-1)$ na qual simplesmente é $\Theta(n)$

8 (b) $n \geq 0$ Considerando n

$r \geq 0$

$T(0) = 1$

$T(n) = T(n-1) + c_1 k + c_2$

Considerando $k = r^n$ $\rightarrow T(n) = T(n-1) + c_1 r^n + c_2$

Pelo metodo de iteração

- expandindo

$T(n-1) = T(n-2) + c_1 r^{n-1} + c_2$

$T(n-2) = T(n-3) + c_1 r^{n-2} + c_2$

- substituindo na formula

$T(n) = T(n-1) + c_1 k^n + c_2$

$= T(n-2) + c_1 r^{n-1} + c_2 + c_1 r^n + c_2$

$= T(n-3) + c_1 r^{n-2} + c_2 + c_1 r^{n-1} + c_2 + c_1 r^n + c_2$

$= T(n-3) + c_1 (r^{n-2} + r^{n-1} + r^n) + 3c_2$

- iesimo passo

$T(n) = T(n-i) + c_1 (\underbrace{r^{n} + r^{n-(i-1)} r^{(n-i-2)}}) + i c_2$

$\rightarrow$ Soma Geometrica

$S = \frac{(1^\circ \text{ termo})(\text{Razão}^n - 1)}{\text{Razão} - 1}$

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

- encontrando i

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

$0 = n - i$

$i = n$

- $T(n) = T(n-n) + c_1 \left(\frac{r(r^n - 1)}{r - 1} \right) + n c_2$

$= T(0) + c_1 \left(\frac{r(r^{n+1} - 1)}{r - 1} \right) + n c_2$

$= \boxed{1 + c_1 \left(\frac{r^{n+1} - 1}{r - 1} \right) + n \cdot c_2}$ Portanto o algoritmo é $O(r^n)$

- Prova por indução

- Base = $T(0) = 1 + c_1 \left(\frac{r - r^0}{r - 1} \right) + 0$

$1 = 1$

- considerando $n-1$ na formula

$T(n-1) = 1 + c_1 \left(\frac{r^n - r}{r - 1} \right) + (n-1) c_2$

- $T(n) = T(n-1) + c_1 r^n + c_2$

$= 1 + c_1 \left(\frac{r^n r}{r - 1} \right) + (n-1) c_2 + c_1 r^n + c_2$

$= 1 + c_1 \left(\frac{r^{n+1} - r}{r - 1} \right) + c_2 n - c_2 + c_2$

$\boxed{= 1 + c_1 \left(\frac{r^{n+1} - r}{r - 1} \right) + c_2 n}$

$\rightarrow$ Como o objetivo é determinar a quantidade de vezes que CAIXA-PRETA é chamada considera-se $c_1 = 1$ e $c_2 = 0$

Assim conclui-se:

$\boxed{1 + \frac{r^{n+1} - r}{r - 1}}$