Exercício 2

Daniel Amorim Vilela de Salis - 123.145

① Realizando a análise do tempo temos:

- Algoritmo I :
  - $T(n) = 5T\left(\frac{n}{2}\right) + \Theta(n)$
  - Podemos utilizar o teorema mestre onde: $a=5, b=2, n^{\log_2 5}$
  - Tomando $\log_2 5$ como $2{,}32$
  - $f(n) = n$ e $\varepsilon = 2{,}32 - 1 \Rightarrow \varepsilon = 1{,}32$
  - Assim vemos que $5 > 2^1$ $(a > b^k)$, dessa forma $T(n) = \Theta(n^{\log_b a})$

    $\therefore T(n) = \Theta(n^{2,32})$

- Algoritmo II:
  - Se comporta da forma $T(n) = 2T(n-1) + \Theta(1)$
  - baseado no método da árvore de recursão temos a tabela:

nível	tamanho	nós	tempo
0	n	1	1
1	$n-1$	2	1
2	$n-2$	4	1
i	$(n-i)$	2^i	1
$n-1$	1	2^n	1

  - O custo do tempo vem do somatório dos custos dos nós, assim temos:

$$T(n) = \sum_{i=0}^{n-1} 2^i \cdot 1 = \left(\frac{2^{n-1+1} - 1}{2-1}\right) = \frac{2^n - 1}{1} = 2^n - 1 \quad \therefore T(n) \text{ será } \Theta(2^n)$$

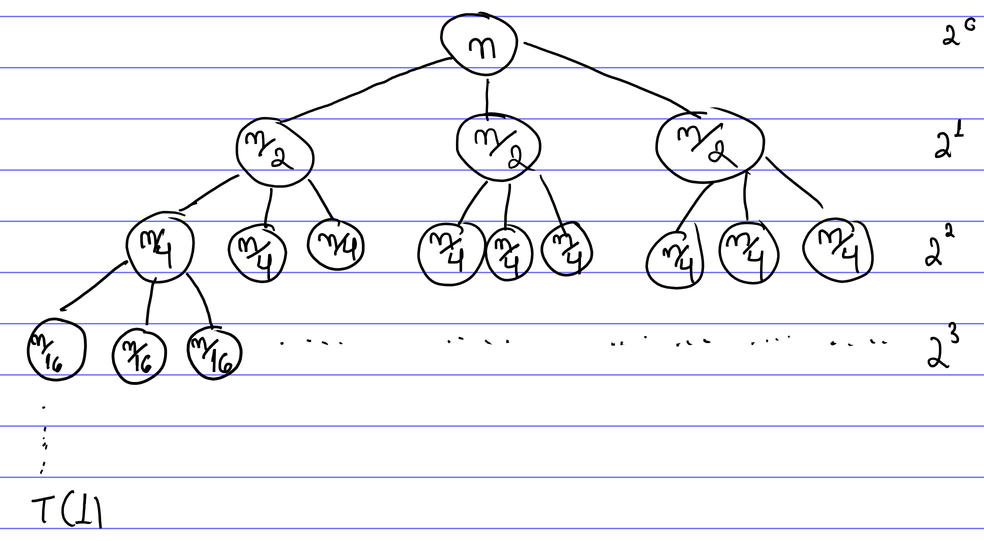
- Algoritmo III:
  - Se comportará da seguinte forma: $T(n) = 9T\left(\frac{n}{3}\right) + \Theta(n^2)$
  - É possível utilizar o teorema mestre
  - $a = 9,\ b = 3,\ n^{\log_3 9} = n^2,\ f(n) = n^2$, assim temos:
  - $\varepsilon = \log_b a - k \Rightarrow \varepsilon = 2 - 2 = 0$
  - $a = b^k \iff 9 = 3^2$
  - Temos portanto o caso 2, temos então:

    $T(n) = \Theta(n^{\log_b a} \cdot \log n) \Rightarrow T(n) = \Theta(n^2 \log n)$

- Escolha do Algoritmo:
  - Podemos descartar o II, pois $\Theta(n^{2,32}) < \Theta(2^n)$ e temos também que $\Theta(n^2 \log n) < \Theta(2^n)$
  - Tomando $f(n) = n^{2,32}$ e $g(n) = n^2 \log n$ teremos que verificar qual terá o menor crescimento
  - Para isso podemos fazer o limite de $\frac{f(x)}{g(x)}$ tendendo ao ∞
  - $\lim\limits_{n \to \infty} \frac{n^{2,32}}{n^2 \log n} = \lim\limits_{n \to \infty} \frac{n^{0,32}}{\log n} \overset{\text{l'hospital}}{=} \lim\limits_{n \to \infty} \frac{0{,}32 \cdot \ln(10) \cdot n^{0,32}}{0{,}68} = \infty$
  - Como $\lim\limits_{n \to \infty} \frac{f(n)}{g(n)} = \infty$, então $f(x)$ cresce mais rápido que $g(x)$
  - Portanto $\Theta(n^{2,32}) > \Theta(n^2 \cdot \log n)$, dessa forma vimos que o Algoritmo III será o melhor para ser escolhido

②

a) $T(n) = 3T\left(\frac{n}{2}\right) + n$



$$T(n) = 3T\left(\frac{n}{2}\right) + n$$

$$T(n) = \sum_{i=0}^{\lg n} 3^i \cdot c\frac{n}{2^i} \Rightarrow T(n) = cn \cdot \sum_{i=0}^{\log n} \left(\frac{3}{2}\right)^i$$

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

$$T(n) = cn \cdot \frac{\left(\frac{3}{2}\right) \cdot \left(\frac{3}{2}\right)^{\lg n} - 1}{\left(\frac{1}{2}\right)}$$

$$T(n) = cn \cdot 2 \cdot \left(\left(\frac{3}{2}\right) \cdot \left(\frac{3}{2}\right)^{\lg n} - 1\right)$$

$$T(n) = cn \cdot 3 \cdot \frac{3^{\lg n}}{2^{\lg n}} - 2 \cdot cn$$

$$T(n) = 3cn \cdot \frac{3^{\lg n}}{\cancel{n^{\lg 2}}\ 1} - 2cn$$

$$T(n) = 3cn \cdot n^{\lg 3} - 2cn$$

$$T(n) = O(n^{\lg 3})$$

④

a) Utilizando uma árvore binária podemos encontrar a solução, onde os andares seriam representados pelos nós. Dessa forma teremos que ir até o 50° andar para jogar o celular, caso quebre o andar máximo estaria entre os andares 1 e 49, caso contrário estará entre o 50 e 100. Se continuarmos executando essa mesma estratégia, veremos que o número de quedas necessárias se dá por $\log_2 x$, sendo x o número de andares. Resolvendo $\log_2 100$ teremos 6,6, arredondando teremos 7 como resposta para o número ótimo de tentativas.

b) Para dois celulares podemos seguir o raciocínio de que caso o primeiro quebre no andar de número n ainda é possível que tenhamos que passar por $(n-1)$ andares anteriores, um à um. Caso o primeiro não quebre podemos subir apenas $(n-1) + n$ andares. Da mesma forma para o próximo andar $n + (n-1) + (n-2)$ e assim por diante. Para este prédio temos: $n + (n-1) + (n-2) \ldots + 1 = 100$. Esse padrão pode ser convertido para a fórmula: $\frac{n(n+1)}{2} = 100 \Rightarrow n \cong 13{,}6$.

Podemos então arredondar para 14 o número de ótimas tentativas.