Resolução <u>resumida</u> do "teste modelo I".

1. (a) F (b) F (c) V (d) V

2. (a) Apresentamos um (contra-)exemplo. Seja $g(n) = n$, $f(n) = n$ para n é par e $f(n) = 2n$ para n é ímpar. Para $n \geq 1$ $f(n)/g(n)$ alterna entre 1 e 2 e por isso não existe $\lim_{n\to\infty} f(n)/g(n)$. Todavia, $f(n) \in \Theta(g(n))$, basta usar $n_0 = 0$, $k_1 = 1$ e $k_2 = 2$ na definição da ordem de grandeza Θ.

   (b) Se fosse, existiam k e n_0 tais que $n^2 \leq k \log n$ para todo o $n \geq n_0$. Supondo $n_0 >= 2$ vem sucessivamente $n \leq k \log n$ e $n/\log n \leq k$ para n suficientemente grande. Mas é fácil ver que isso é contraditório com $\lim_{n\to\infty} n/\log n = +\infty$.

3. Usando um resultado no fundo da folha, usamos a equação característica $(r-1)(r-3) = 0$ e portanto a solução geral é $f(n) = a \times 1^n + b \times 3^n = a + b \times 3^n$. Do sistema de equações

$$\begin{cases} a+b &= 0 \\ a+3b &= 3 \end{cases}$$

obtém-se $a = -3/2$, $b = 3/2$ e a solução é $f(n) = 3/2(3^n - 1)$.

4.
```
Tabelar  n   f(n) | Suspeitar:   f(n) = n(n-1)
        ----------| Demonstrar:
         0     0  |   Caso base n=0: f(n) = 0 da recorrência,
         1     0  |     e f(n)=0 da fórmula n(n-1)
         2     2  |    Passo indutivo,  f(n)=n(n-1) ==> f(n+1) = (n+1)n
         3     6  |      f(n+1) = f(n) + 2n   (da recorrência)
         4    12  |             = n(n-1)+2n   (hipótese indutiva)
         5    20  |             = n^2+n = (n+1)n
```

5. Para constantes- c', c''e c apropriadas vem

$$\begin{cases} t(1) &= c' \\ t(2n) &= 6t(n) + c''n^2 \end{cases} \qquad \begin{cases} t(1) &= c \\ t(n) &= 6t(n/2) + cn^2 \end{cases}$$

Usando um resultado no fundo da folha temos $a = 6$, $b = 2$, $k = 2$, sendo $a > b^k$. Logo $t(n)$ tem ordem $\Theta(n^{log 6})$.

6. É após os redimensionamentos que $\sum c_i/n$ é máximo, Quando há um redimensionamento, $n = 4^p+1$, sendo o custo total dos redimensionamentos (este mais os anteriores) $(n-1)+(n-1)/4+(n-1)/16+\ldots+1 \leq 4/3(n-1)$. Assim, o custo total não excede $n+4(n-1)/3 \leq 7n/3$. Podemos então usar como custo amortizado $7/3$, pois para todo o n é $7n/3 \geq \sum_{i=1}^{n} c_i$.

   Consideremos um `PUSH` em que o vector é redimensionado (há uma infinidade destes `PUSH`'s) e suponhamos que nesta operação n (número de elementos no "stack") passa para $n+1$. A soma dos custos até este `PUSH` é, pelo menos, $n+1$ (`PUSH`'s anteriores mais este) $+$ n (este redimensionamento). Assim, o custo amortizado não pode ser inferior a $(2n+1)/(n+1) = 2 - 1/(n+1)$. O custo amortizado, sendo uma constante, tem que ser pelo menos 2. Note-se que este resultado é válido para qualquer regra (não necessariamente multiplicar por uma constante) de aumento do tamanho do "stack"