ECM-306 -Tópicos Avançados em Estrutura de Dados
Prof. Calvetti - Exercícios Big-Oh
Análise Assintótica de Algoritmos

1. Um algoritmo apresenta Função de Complexidade $\mathbf{F(n) = 5n^2 + 10n + 8}$. Prove ou disprove que a Função de Complexidade $\mathbf{F(n)}$ pertence à ordem de complexidade $\mathbf{O(n^2)}$.

$$F(n) = 5n^2 + 10n + 8 \leq 5n^2 + 10n^2 + 8n^2$$

$$\leq 23 \cdot n^2$$

$\therefore \exists\ c = 23,\ c > 0$ tal que para todo $n \geq 1$, tem-se:

$$5n^2 + 10n + 8 \leq \underbrace{23}_{c} \cdot n^2$$

$\therefore\ 5n^2 + 10n + 8$ é $O(n^2)$

ECM-306 -Tópicos Avançados em Estrutura de Dados
Prof. Calvetti - Exercícios Big-Oh
Análise Assintótica de Algoritmos

2. Um algoritmo apresenta Função de Complexidade **$F(n) = n^3$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n^3)$.**

$$F(n) = 5n^2 + 10n + 8 \leqslant 5n^3 + 10n^3 + 8n^3$$
$$\leqslant 23n^3$$

$\therefore \exists\ c = 23,\ c > 0$ tal que

para todo $n \geqslant 1$, tem-se

$$5n^2 + 10n + 8 \leqslant \underbrace{23}_{c} \cdot n^3$$

$\therefore\ 5n^2 + 10n + 8$ é $O(n^3)$

3. Um algoritmo apresenta Função de Complexidade **$F(n) = 5n^2 + 10n + 8$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n)$.**

$$F(n) = 5n^2 + 10n + 8 \leq c \cdot g(n)$$

$$5n^2 + 10n + 8 \leq c \cdot n$$

$$5n^2 + 10n + 8 - cn \leq 0$$

$$n \cdot (5n + 10 - c) \leq -8$$

n é sempre positivo e suficientemente grande.

$$\therefore \quad 5n + 10 - c < 0$$

$$\therefore \quad 5n < c - 10$$

$$\therefore \quad n < \underbrace{\frac{c-10}{5}}_{K} \Rightarrow n < K$$

Absurdo: n é suficientemente grande e, portanto, não pode ser menor que uma constante.

Como n cresce indefinidamente, para algum n_0 $(n > n_0)$, n ultrapassará a constante K.

Portanto, $5n^2 + 10n + 8$ NÃO é $O(n)$

4. Um algoritmo apresenta Função de Complexidade **$F(n) = n^3$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n^2)$.**

$F(n) = n^3$ é $O(n^2)$?

$n^3 \leq c \cdot g(n)$

$n^3 \leq c \cdot n^2$

$\frac{n^3}{n^2} \leq c \quad \therefore \quad \boxed{n \leq c}$

Absurdo, pois $n > 0$ e suficientemente grande!

$\therefore$ n não pode ser menor do que uma constante!

Como n cresce indefinidamente, sempre para algum n_0, n ultrapassará c.

$\therefore$ $F(n) = n^3$ <u>NÃO</u> é $O(n^2)$.

5. Um algoritmo apresenta Função de Complexidade **F(n) = n**. Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n).**

$F(n) = n$ é $O(n)$?

$n \leq c \cdot g(n)$

$n \leq c \cdot n$

$\therefore$ p/ $c = 1$, tem-se $n \leq 1 \cdot n$, p $\forall n > 1$

$\therefore$ $F(n)$ é $O(n)$

6. Um algoritmo apresenta Função de Complexidade **$F(n) = 50n^2$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n)$.**

$F(n) = 50n^2$ é $O(n)$?

$$50n^2 \leq c \cdot g(n)$$

$$50n^2 \leq c \cdot n$$

$$\therefore \quad 50n \leq c$$

$$\therefore \quad n \leq \frac{c}{50} = K \text{ (constante)}$$

Absurdo! n não pode ser menor do que uma constante $K>0$, pois n cresce indefinidamente $(n \rightarrow \infty)$

$\therefore$ $50n^2$ não é $O(n)$

7. Um algoritmo apresenta Função de Complexidade **$F(n) = 50n^2$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n^3)$.**

$F(n) = 50n^2$ é $O(n^3)$?

$$50n^2 \leq c \cdot g(n)$$
$$50n^2 \leq c \cdot n^3$$
$$50n^2 \leq 50 \cdot n^2 \text{ , p/ } \forall n > 0$$

$\therefore \exists c = 50, c > 0, \mid 50n^2 \leq 50n^3, \forall n > 0$

Logo $50n^2$ é $O(n^3)$

8. Um algoritmo apresenta Função de Complexidade **$F(n) = n^2 - 200n - 300$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n)$.**

$F(n) = n^2 - 200n - 300$ é $O(n)$?

$$n^2 - 200n - 300 \leq c \cdot g(n)$$
$$n^2 - 200n - 300 \leq c \cdot n$$
$$n^2 - 200n - cn \leq 300$$
$$n \cdot (n - 200 - c) \leq 300$$
$$n - 200 - c \leq \frac{300}{n}$$

Para n muito grande, $\frac{300}{n}$ tende a zero!

$\therefore$ $n - 200 - c \leq 0$, p/ n grande

$\therefore$ $n \leq \underbrace{200 + c}_{K}$, $c > 0$

Absurdo, pois n é grande e não pode ser menor que uma constante K.
Em algum $n > n_0$ $(n_0 \geq 1)$, n ultrapassa K.

Logo, $n^2 - 200n - 300$ não é $O(n)$

9. Um algoritmo apresenta Função de Complexidade **F(n) = n^2 – 200n – 300**. Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n^5).**

$F(n) = n^2 - 200n - 300$ é $O(n^5)$?

$$n^2 - 200n - 300 \leq c \cdot g(n)$$

$$n^2 - \cancel{200n} - \cancel{300} \leq c \cdot n^5$$

$$\therefore \quad n^2 \leq c \cdot n^5$$

$$\exists\, c = 1,\ c > 0,\ |\ n^2 \leq 1 \cdot n^5$$

Portanto, $n^2 - 200n - 300 \leq 1 \cdot n^5$

Logo $n^2 - 200n - 300$ é $O(n^5)$

10. Um algoritmo apresenta Função de Complexidade **F(n) = n^2 – 200n – 300**. Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(1).**

$F(n) = n^2 - 200n - 300$ é $O(1)$?

$n^2 - 200n - 300 \leq c \cdot g(n)$

$n^2 - 200n - 300 \leq c \cdot 1$

$n^2 - 200n \leq \underbrace{c + 300}_{K_1}$, $K_1 > 0$

$\therefore\ n^2 - 200n \leq K_1$

$\therefore\ n(n - 200) \leq K_1 \Rightarrow n - 200 \leq \frac{K_1}{n} \rightarrow 0$

p/n grande, $\frac{K_1}{n}$ tende a zero!

$\therefore\ n - 200 \leq 0 \Rightarrow n \leq \underbrace{200}_{K_2}$

Absurdo, para n suficientemente grande, n não pode ser menor que K_2.

Logo, $n^2 - 200n - 300$ <u>NÃO</u> é $O(1)$

11. Um algoritmo apresenta Função de Complexidade **F(n) = 3/2n² +7/2n – 4**. Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n).**

$$F(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4 \text{ é } O(n)?$$

$$\frac{3}{2}n^2 + \frac{7}{2}n - 4 \leq c \cdot n$$

$$\frac{3}{2}n^2 + \frac{7}{2}n - cn \leq 4$$

$$n \cdot \left(\frac{3}{2}n + \frac{7}{2} - c\right) \leq 4$$

$$\frac{3}{2}n + \frac{7}{2} - c \leq \frac{4}{n}$$

tende a zero p/n muito grande

$$\therefore \frac{3}{2}n + \frac{7}{2} - c \leq 0 \text{ , p/} n \to \infty$$

$$\frac{3}{2}n \leq \underbrace{c - \frac{7}{2}}_{K_1} \to \frac{3}{2}n \leq K_1 \Rightarrow n \leq \underbrace{\frac{2K_1}{3}}_{K_2}$$

$\therefore n \leq K_2$; ABSURDO! pois n é muito grande!

Logo: $\frac{3}{2}n^2 + \frac{7}{2}n - 4$ NÃO é $O(n)$!

12. Um algoritmo apresenta Função de Complexidade **$F(n) = 3/2n^2 + 7/2n - 4$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n^2)$.**

$$F(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4 \text{ é } O(n^2) \text{ ?}$$

$$\begin{aligned}
\frac{3}{2}n^2 + \frac{7}{2}n \cancel{-4} &\leq c \cdot n^2 \\
&\leq \frac{3}{2}n^2 + \frac{7}{2}n \\
&\leq \frac{3}{2}n^2 + \frac{7}{2}n^2 \\
&\leq \frac{10}{2}n^2 \\
&\leq 5n^2
\end{aligned}$$

$\therefore \exists\ K = 5,\ K > 0$, tal que

$$\frac{3}{2}n^2 + \frac{7}{2}n - 4 \leq 5 \cdot n^2, \quad \forall n \geq 1$$

Logo: $\frac{3}{2}n^2 + \frac{7}{2}n - 4$ é $O(n^2)$

13. Um algoritmo apresenta Função de Complexidade **$F(n) = 5n^2 + 10n - 900$**. Prove ou disprove que a Função de Complexidade **$F(n)$** pertence à ordem de complexidade **$O(n^2)$.**

$F(n) = 5n^2 + 10n - 900$ é $O(n^2)$?

$$5n^2 + 10n \cancel{-900} \leq c \cdot n^2$$

$$\leq 5n^2 + 10n$$

$$\leq 5n^2 + 10n^2$$

$$\leq 15n^2$$

$\therefore \exists c, c > 0 \ (c = 15) \mid 5n^2 + 10n - 900 \leq 15 \cdot n^2$

Logo $5n^2 + 10n - 900$ é $O(n^2)$

14. Um algoritmo apresenta Função de Complexidade **F(n) = 5n² +10n – 900**. Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n).**

$F(n) = 5n^2 + 10n - 900$ é $O(n)$?

$$5n^2 + 10n - 900 \leq c \cdot n$$

$$5n^2 + 10n - cn \leq 900$$

$$\therefore n \cdot (5n + 10 - c) \leq 900$$

$$5n + 10 - c \leq \frac{900}{n} \to 0$$

tende a zero quando $n \to \infty$

$\therefore 5n + 10 - c \leq 0$, p/ n grande

$$\therefore 5n \leq c - 10 \to n \leq \underbrace{\frac{c-10}{5}}_{K}$$

Absurdo, n sempre irá ultrapassar uma constante p/ algum n_0.

Logo $5n^2 + 10n - 900$ não é $O(n)$

15. Um algoritmo apresenta Função de Complexidade **F(n) = 10 + 2/n**. Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n²).**

$F(n) = 10 + \frac{2}{n}$ é $O(n^2)$?

$$10 + \frac{2}{n} \leq c \cdot n^2$$

p/ n grande, $\frac{2}{n}$ tende a zero

Portanto, $10 + \frac{2}{n} \leq 10 \cdot c^2$, $n > 0$

logo $\exists\ c = 10,\ c > 0,\ |\ 10 + \frac{2}{n} \leq 10 \cdot n^2$

$\therefore\ 10 + \frac{2}{n}$ é $O(n^2)$