MO417 — Complexidade de Algoritmos I — 2s2019

Lista de Exercício 2

Além dos exercícios abaixo, recomendo que façam a maior quantidade possível de exercícios dos livros texto (CLRS e Manber) dos capítulos relacionados.

Questão 1. Sejam k um inteiro positivo e p um polinômio de grau k.

1. Prove que $p = O(n^k)$.
2. Prove que se o coeficiente de n^k em p for positivo, então $p = \Theta(n^k)$.

Questão 2. Prove que:

1. $2^n = O(3^n)$
2. $\log_{10} n = \Theta(\log_2 n)$
3. $n^k = O(n^{k+1})$, para todo $k \geq 0$
4. $n^k = o(n^{k+1})$, para todo $k \geq 0$
5. $n^k = o(n^{k+\varepsilon})$, para todo $k \geq 0$ e todo $\varepsilon > 0$
6. $3^n = \Omega(2^n)$
7. $3^n = \omega(2^n)$
8. $n = \omega(\lg n)$
9. $\lg\lg n = o(\lg n)$
10. $n^k = o(n^{\lg n})$
11. $\lg^2 n = o(n)$
12. $n = \omega(\lg^k n)$, para todo $k \geq 0$
13. $n^{\lg n} = o(2^n)$
14. $a^n = \omega(n^{\lg n})$, para todo $a > 1$
15. $n! = \omega(2^n)$ (dica: ache um limitante inferior para $n!$ e compare com 2^n)
16. $n! = O(n^n)$

Questão 3. Resolva as seguintes recorrências (sem usar o Teorema Mestre), considerando $T(1) = 1$. Você pode dar a fórmula usando notação assintótica. Quando o Teorema Mestre se aplicar, diga qual dos três casos se aplica e confira a sua resposta com a do Teorema Mestre.

1. $T(n) = T(n-1) + 1$
2. $T(n) = T(n-1) + n$
3. $T(n) = T(n/2) + 1$
4. $T(n) = T(n/2) + n$
5. $T(n) = 2T(n-1) + 1$
6. $T(n) = 2T(n/2) + 1$
7. $T(n) = 2T(n/2) + \lg n$
8. $T(n) = 2T(n/2) + n$
9. $T(n) = 2T(n/2) + n\lg n$
10. $T(n) = 2T(n/2) + n^2$

11. $T(n) = 3T(n/3) + n$
12. $T(n) = 2T(n/3) + 1$
13. $T(n) = 2T(n/3) + n$
14. $T(n) = 4T(n/3) + n$

Questão 4. Apresente um pseudocódigo utilizando a sintaxe vista em aula (do CLRS) para o algoritmo da Busca-Binária. Mostre que o algoritmo para para toda entrada e mostre qual é o tempo do melhor e do pior caso do algoritmo.