Lista1

Pedro Correa

1. O que significa que um algoritmo tem complexidade $O(1)$? De três exemplos dealgoritmos que seguem esta ordem de complexidade

Dizemos que quando o algoritmo possui uma complexidade de $O(1)$ dizemos por que o tempo para ele ser executado eh de apenas uma linha de codigo a ser executada. Tornando assim bem rapido sua execucao

Exemplos de coisas que possui complexidade $O(1)$:

- Acessar um item em um array
- Adicionar um item ao inicio de uma lista ligada

---

2. Dois algoritmos diferentes têm números de operações diferentes $20n\sqrt{n}$ e n^2. Qual dos dois algoritmos é mais eficiente para um problema de tamanho n? Qual é o melhor algoritmo para problemas pequenos (digamos, $n < 10$)? Qual é o menor problema em que o primeiro algoritmo vence o segundo?

Digamos que o o primeiro algoritmos é melhor para complexidade de **n**, pois somente devemos olhar a portencia que estamos trabalhando. Como o segundo algoritmo é elevado ao quadrado, ele teria muito mais complexidade que o primeiro algoritmo que possui a complexidade de $O(n\sqrt{n})$

Mas para numeros pequeno, no caso de quando o **n** for menor que 10, o segundo algoritmo possui uma menor complexidade e o primeiro possui uma maior complexidade. Isso ocorre pois no primeiro estamos multiplicando por 20 e pela $\sqrt{n}$, o que quando o valor de **n** é muito baixo acaba tornando uma complexidade maior que apenas uma complexidade ao quadrada.

A quantidade que **n** deve ter para vencer o segundo algoritmo é **401**

---

3. Dois algoritmos diferentes têm funções de complexidade n^2 e 2^n para resolver um problema de tamanho **n**. Qual algoritmo é mais eficiente? Para que tamanhos do problemaos dois algoritmos têm o mesmo desempenho? Qual é o menor problema em que o primeiro algoritmo vence o segundo?

O primeiro algoritmo é mais eficiente pois ele nao é exponencial conforme o tamanho igual ao segundo algoritmo

Os valores foram iguais somente quando **n** foi 2 e 4

O menor valor para o primeiro algorimo vencer é **1**

---

4. Prove por definição as seguintes afirmações:

a. $n^2 = O(n^3)$

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

$$1 \leq c * n$$

$$c = 1; n = 2$$

b. $2 * n^2 + 1 = O(n^2)$

$$2 * n^2 + 1 \leq c * n^2$$

$$2 * n^2 - c * n^2 \leq -1$$

$$n^2 * (2 - c) \leq -1$$

$$c > 2$$

c. $n^2 + 3 * n + 7 = \Omega(6 * n + 7)$

$$n^2 + 3 * n + 7 \geq c * (6 * n + 7)$$

$$n^2 + 3 * n + 7 \geq 6 * c * n + 7 * c$$

$$n^2 + 3 * n - 6 * c * n - 7 * c \geq -7$$

$$n(n + 3 - 6 * c) - 7 * c \geq -7$$

$$c = \frac{1}{2}; n \geq 2$$

d. $1000 * n = O(\frac{n^2}{1000})$

$$1000 * n \geq c * (\frac{n^2}{1000})$$

$$1000 \geq \frac{c * n}{1000}$$

$$1000 * 1000 \geq c * n$$

$$10^6 \geq c * n$$

e. $\frac{1}{2} * n * (n + 1) = \Theta(n^2)$

$$c1 * n^2 \leq \frac{1}{2} * n * (n + 1)$$

$$c1 * n^2 \leq \frac{1}{2} * n^2 + \frac{1}{2} * n$$

$$c1 * n \leq \frac{1}{2} * n + \frac{1}{2}$$

$$c1 * n - \frac{1}{2} * n \leq \frac{1}{2}$$

$$c1 = 1; n \leq 1$$

$$\frac{1}{2} * n * (n + 1) \leq c2 * n^2$$

$$\frac{1}{2} * n^2 + \frac{1}{2} * n \leq c2 * n^2$$

$$\frac{1}{2} * n + \frac{1}{2} \leq c2 * n$$

$$\frac{1}{2} \leq c2 * n - \frac{1}{2} * n$$

$$c2 = 1; n > 1$$

f. $\log(n^2) = \Omega(\log(n^2))$

$$\log(n^2) \geq c * \log(n^2)$$

$$2 * \log(n) \geq 2 * c * \log(n)$$

$$1 \geq c$$

g. $10 * n^2 + 12 * n + 6 = \Theta(2 * n^2 - n)$

$$10 * n^2 + 12 * n + 6 \leq c1 * (2 * n^2 - n)$$

$$10 * n^2 + 12 * n + 6 \leq 2 * c1 * n^2 - c1 * n$$

$$10 * n^2 - 2 * c1 * n^2 + 12 * n + c1 * n \leq -6$$

$$n * (10 * n - 2 * c1 * n + 12 + c1) \leq -6$$

$$c1 * (2 * n^2 - n) \leq 10 * n^2 + 12 * n + 6$$

$$2 * c1 * n^2 - c1 * n \leq 10 * n^2 + 12 * n + 6$$

$$-6 \leq 10 * n^2 - 2 * c1 * n^2 + 12 * n + c1 * n$$

$$-6 \le n * (10 * n - 2 * c1 * n + 12 + c1)$$

h. $2^{n+1} = O(2^n)$

$$2^{n+1} \le c * (2^n)$$

$$\frac{2^{n+1}}{2^n} \le c$$

$$2^{n+1-n} \le c$$

$$2 \le c$$

i. $2^{2*n} = O(2^n)$

$$2^{2*n} \le c * 2^n$$

$$2^2 \le c$$

$$4 \le c$$