Lista de Exercícios 01

Q1

50 = 10 s
51 = 20 s
52 = 40 s
53 = 80 s
54 = 160 s
55 = 320 s
56 = 640 s
57 = 1280 s
58 = 2560 s
59 = 5120 s
60 = $10*2^{10}$ s
...
70 = $10*2^{20}$ s
...
80 = $10*2^{30}$ s
...
90 = $10*2^{40}$ s
...
100 = $10*2^{50}$ s

Q2

a) Ordena os valores do vetor, colocando-os em ordem crescente, mas apenas nos índices de ‘1’ até ‘n-1’.

b) Pior caso é quando o vetor está em ordem decrescente e ‘n’ seja igual a (tamanho do vetor + 1). Seria necessário realizar a troca em todas as iterações e em todo o vetor, mas ainda sem o índice ‘0’.

Complexidade $O(n^2)$, pois tem um laço 'for' dentro de outro e essas iterações possuem tamanhos diferentes.

Q3

Não. O algoritmo serve para solucionar problemas. Ignorar a eficiência pode criar mais problemas, dentre eles, estouro de memória e/ou lentidão ao obter a saída (que seria muito ruim caso o algoritmo fornecesse uma saída, 100 anos depois de ter iniciado).

Q4

Não. $n^2 / 2$, possui polinômio de grau 2 que domina polinômios de graus menores. O correto seria $O(n^2)$, pois a elevação ao quadrado, aumenta significativamente o número de casos em relação a ‘n’, somente.

Q5

b

Q6

f(n) é $O(n^2)$, para todo polinômio que contenha pelo menos um ‘$a*n^2$’ e esse ‘a’ deve ser maior que 0 ($a > 0$). Nesse caso ‘$n^2 – 100n$’ é ‘$O(n^2)$’ pois ‘a’ está oculto e subentendido como ‘a = 1’, que contempla a propriedade ‘$a > 0$’, e não porque ‘$n >= 100$’. Então, é incorreta a afirmação ‘$n^2 – 100n$ está em $O(n^2)$ para todo $n >= 100$’. O correto seria, ‘$n^2 – 100n$ é $O(n^2)$ pois $a > 0$ em $a*n^2$’.

Q7

- Propriedade I: f(n) é O(f(n))
  - $f(n^{0,5})$ é $O(n^{0,5})$
  - $f(n^{0,5}) = lgn^{0,5} = 0,5 * lgn => 0,5 * f(n^{0,5})$

- Propriedade II: c * O(f(n)) é O(f(n))
  - Se f(n) é O(f(n)) e c * O(f(n)) é O(f(n)), então c * O(f(n)) é c * f(n)
  - $c * f(n) = 0,5 * f(n^{0,5})$
  - $0,5 * O(n^{0,5}) = O(n^{0,5})$

Q8

f(n) = lgn

Para que f(n) seja Ω(g(n)), deve-se satisfazer a seguinte condição:

‘f(n) >= cg(n)’, onde ‘c > 0’ e n >= n0 >= 1

A condição é satisfeita para n = 1 e c > 0 (‘c’ com qualquer valor de 1 até infinito):

f(n) = lg1 = 0 g(n) = lg1 = 0

c = 2 2 * lg1 = $lg1^2$ = 0

c = 3 3 * lg1 = $lg1^3$ = 0

…

Nesse caso, ‘f(n) = cg(n)’ para n = 1 e c > 0.

Mas a condição não é satisfeita para n > 1 e c > 0:

Para n = 2:

f(n) = lg2 = 0,69 g(n) = lg2 = 0,69

c = 2 2 * lg2 = $lg2^2$ = 1,38

c = 3 3 * lg2 = $lg2^3$ = 2,07

...

Nesse caso, ‘f(n) < cg(n)’ para n = 2 e c > 0. Portanto, lgn não está em Ω(n).

Q9

Para f(n) = n(n + 1) / 2

Para que f(n) seja $\Theta(n)$, deve-se satisfazer as seguintes condições:
‘c1 * g(n) <= f(n) <= c2 * g(n)’, onde ‘c1 > 0 e c2 > 0’ e n >= n0 >= 1

A condição é satisfeita para n = 1 e c1 > 0 e c2 > 0:
f(n) = 1(1 + 1) / 2 = 1 g(n) = 1(1 + 1) / 2 = 1
c1 = 0,5 0,5(1(1 + 1) / 2) = 0,5 c2 = 3 3(1(1 + 1) / 2) = 3
c1 = 0,7 0,7(1(1 + 1) / 2) = 0,7 c2 = 4 4(1(1 + 1) / 2) = 4
…
A condição é satisfeita para n = 2 e c1 > 0 e c2 > 0:
f(n) = 2(2 + 1) / 2 = 3 g(n) = 2(2 + 1) / 2 = 3
c1 = 0,5 0,5(2(2 + 1) / 2) = 1,5 c2 = 3 3(2(2 + 1) / 2) = 9
c1 = 0,7 0,7(2(2 + 1) / 2) = 2,1 c2 = 4 4(2(2 + 1) / 2) = 12
…
Nesses casos, ‘c1 * g(n) <= f(n) <= c2 * g(n)’ para n >= 1 e c1 > 0 e c2 > 0.

Para f(n) = n(n – 1) / 2

Para que f(n) seja $\Theta(n)$, deve-se satisfazer as seguintes condições:
‘c1 * g(n) <= f(n) <= c2 * g(n)’, onde ‘c1 > 0 e c2 > 0’ e n >= n0 >= 1

A condição é satisfeita para n = 1 e c1 > 0 e c2 > 0:
f(n) = 1(1 – 1) / 2 = 0 g(n) = 1(1 – 1) / 2 = 0
c1 = 0,5 0,5(1(1 – 1) / 2) = 0 c2 = 3 3(1(1 – 1) / 2) = 0
c1 = 0,5 0,5(1(1 – 1) / 2) = 0 c2 = 4 4(1(1 – 1) / 2) = 0
…
A condição é satisfeita para n = 2 e c1 > 0 e c2 > 0:
f(n) = 2(2 – 1) / 2 = 1 g(n) = 2(2 – 1) / 2 = 1
c1 = 0,5 0,5(2(2 – 1) / 2 = 0,5 c2 = 3 3(2(2 – 1) / 2 = 3
c1 = 0,7 0,7(2(2 – 1) / 2 = 0,7 c2 = 4 4(2(2 – 1) / 2 = 4
…
Nesses casos, ‘c1 * g(n) <= f(n) <= c2 * g(n)’ para n >= 1 e c1 > 0 e c2 > 0.

Existe também a regra de dominância do grau de polinômio, onde um número n^x domina n^x-1. Nesse caso, ao executar a multiplicação da função n(n + 1) / 2 e n(n – 2) /2, temos $\frac{n^2 + n}{2}$ e $\frac{n^2 - n}{2}$. Então as constantes saem, ficando apenas com a notação $n^2 + n$. Como ‘n^2’ domina ‘n’, então sobra apenas n^2. Portanto, n(n + 1) / 2 e n(n – 1) / 2 estão em $\Theta(n^2)$.

Q10

d

Q11

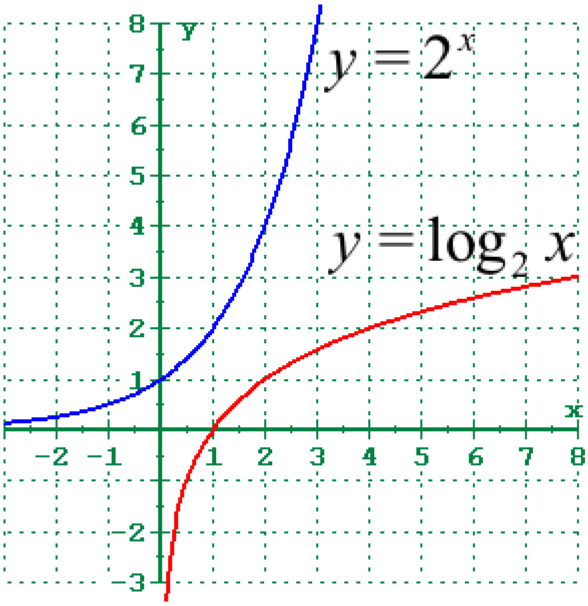
b

Q12

Descobriram que o ponto n0 é 100. A partir desse ponto, um algoritmo domina, assintoticamente, o outro. Sendo assim, o algoritmo f(nlgn) é $O(c*n^2)$ para n >= 100 e c > 0.

Q13

Casos 1 e 2, pois o gráfico logarítmico se estabiliza mais em relação ao eixo y (tem pouca variação com o passar do tempo), enquanto que o gráfico n^2 e n^3 são mais exponenciais.



Q14

b