

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO RN
GERÊNCIA DE TECNOLOGIA DA INFORMAÇÃO E EDUCACIONAL DE TELEMÁTICA
Exercícios de Estrutura de Dados I

1. O algoritmo A usa $100n$ operações enquanto o algoritmo B usa $3n^2$ operações. Determine n_0 para o qual A é melhor do que B para $n \geq n_0$.
2. Um algoritmo leva 2ms para terminar com uma entrada de tamanho 100. Qual o tamanho da entrada para que o algoritmo termine em 10, 30 e 60 segundos quando a ordem de complexidade for:
 - (a) $O(n)$
 - (b) $O(n^2)$
 - (c) $O(\log n)$
 - (d) $O(2^n)$
3. Um algoritmo leva 0.5ms para terminar com uma entrada de tamanho 100. Em quanto tempo o algoritmo termina quando a entrada for de tamanho 10.000 e a ordem de complexidade for:
 - (a) $O(n)$
 - (b) $O(n^2)$
 - (c) $O(\log n)$
 - (d) $O(2^n)$
4. Ordene as funções a seguir pela ordem de complexidade: n^2 , n , $\log n$, 2^n , $n \log n$, n^3 .
5. Mostre que 2^{n+2} é $O(2^n)$
6. Faça
 - (a) Considere A um arranjo (*array*) ordenado de n números inteiros. Desenvolva um algoritmo que receba A e um número inteiro n e retorne a posição do arranjo em que n se encontra.
 - (b) Qual o número de operações do seu algoritmo no pior caso? e no melhor?
 - (c) Qual a ordem de complexidade do algoritmo? Este algoritmo é bom?
7. O que faz o seguinte algoritmo? Analise seu tempo de execução do pior caso e expresse seu valor usando a notação “big-Oh”.

Algorithm $A(a, n)$:
 Entrada: dois inteiros, a e n
 Saída: ?
 $k \leftarrow 0$
 $b \leftarrow 1$
 enquanto $(k < n)$ **faça**
 $k \leftarrow k + 1$
 $b \leftarrow b * a$
 retorne b

8. O que faz o seguinte algoritmo? Analise seu tempo de execução do pior caso e expresse seu valor usando a notação “big-Oh”.

Algoritmo B (a, n):

Entrada: dois inteiros, a e n

Saída: ?

$k \leftarrow n$

$b \leftarrow 1$

$c \leftarrow a$

enquanto ($k > 0$) **faça**

se ($k \% 2 = 0$) **então**

$k \leftarrow k/2$

$c \leftarrow c * c$

senão

$k \leftarrow k - 1$

$b \leftarrow b * c$

retorne b

9. O que faz o seguinte algoritmo? Analise seu tempo de execução do pior caso e expresse seu valor usando a notação “big-Oh”.

Algoritmo C (A, t, x):

Entrada: Um *array* A (ordenado) e dois inteiros t e x

Saída: ?

$l \leftarrow 0$

$r \leftarrow t - 1$

$ac \leftarrow 0$

enquanto ($(l \leq r) \wedge (ac = 0)$) **faça**

$k \leftarrow (l + r)/2$

se ($x = A[k]$) **então**

$ac \leftarrow 1$

senão se ($x < A[k]$) **então**

$r \leftarrow k - 1$

senão

$l \leftarrow k + 1$

se ($ac = 1$) **então**

retorne k

senão

retorne -1

10. Seja A um *array* de inteiros (positivos e negativos), com tamanho n . Calcular a **maior subsequência** de A . A **maior subsequência** é a sequência cuja soma possui o maior valor.

Por exemplo, considere $A = \{-2, 11, -4, 13, -5, 2\}$, a sequência que possui o maior valor é a que corresponde a do segundo ($A[1]$) ao quarto ($A[3]$) elemento $\{11, -4, 13\}$, cuja soma é 20. Para a sequência $A = \{1, -3, 4, -2, -1, 6\}$ a maior é a que corresponde do terceiro ($A[2]$) ao último elemento ($A[5]$), cuja soma é 7. Se todos os elementos forem negativos, a **maior subsequência** possui zero elementos e sua soma é 0.

Desenvolva um algoritmo que calcule a maior subsequência.