

## Lista de exercícios para a AV1

1 - Conte o número de passos e calcule a complexidade  $O(n)$  dos trechos de código abaixo:

- (a) `int sum = 0;`  
`for(int i=0; i<=n; i++ )`  
`sum++;`
- (b) `int sum = 0;`  
`for( int i=1; i<n; i++ )`  
`for( int j=1; j<n; j++ )`  
`sum++;`
- (c) `int sum = 0;`  
`for( int i=0; i<=n*n; i++ )`  
`for( int j=0; j<n*n; j++ )`  
`sum++;`
- (d) `int sum = 0;`  
`for( int i=0; i<n; i++ )`  
`for( int j=1; j<= n; j++ )`  
`sum++;`

2 - Coloque em ordem crescente as seguintes complexidades:

- a)  $2^n, n!, n^5, 10000, e \log(n)$ .
- b)  $n \log(n), n + n^2 + n^3, 2^4 e \sqrt{n}$ .

3 - Obtenha o valor da notação 'O' das seguintes expressões de complexidade:

- a)  $5n^{5/2} + n^{2/5}$
- b)  $6 \log(n) + 9n$
- c)  $3n^4 + n \log(n)$
- d)  $5n^2 + n^{3/2}$

4 - Considere os seguintes algoritmos abaixo:

```
prog( int x )
  se ( x < 5 )
    retorna ( 3*x )
  senão
    retorna(2 * prog(x - 5) + 7)
  fim se
```

```
prog( int x, int y )
  se ( x < y )
    retorna ( -3 )
  senão
    retorna (prog(x - y, y + 3) + y)
  fim se
```

O que é retornado nas seguintes chamadas?

- a) prog (2, 7)
- b) prog (5, 3)
- c) prog (15, 3)

5) Seja  $v$  um vetor de inteiros positivos. Escreva *métodos recursivos* para:

- a) calcular a soma dos elementos em  $v$ ;
- b) calcular o valor mínimo;
- c) calcular o valor máximo;
- d) calcular a média dos valores em  $v$ ;
- e) imprimir os valores do vetor em ordem inversa (sem trocá-los de ordem no vetor)

6) Dado o vetor  $X=[8, 4, 1, 0, 6, 5, 2, 9, 3, 7]$ , ordená-lo, detalhadamente, pelos métodos

- a) QuickSort
- b) MergeSort

7) Determine a notação assintótica  $O$  para as funções a seguir:

- a)  $(n^2-n)/2$
- b)  $n + 2n^{1/2}$
- c)  $n^2 + 3n + 4$
- d)  $2(\log(n))^2$

8) Determine a notação assintótica  $O$  para as funções a seguir:

- a)  $f_1(x) = 3x^2 + 2x + 5$
- b)  $f_2(x) = 4x^2 + 2y + 5x$
- c)  $f_3(x) = 2x + \log x + 1$
- d)  $f_4(x) = 542$
- e)  $f_5(x) = x^4 - 3x + 3$
- f)  $f_6(x) = 14$
- g)  $f_7(x) = n(n-1)/2$

9) Qual dos algoritmos abaixo não utiliza o método de dividir para conquistar?

- a) quickSort
- b) mergeSort
- c) heapSort

10) Qual dos algoritmos abaixo pode utilizar uma estrutura de dados, do tipo árvore binária, para solucionar o problema de ordenação?

- a) quickSort
- b) mergeSort
- c) heapSort

11) Qual dos algoritmos abaixo combina sub-listas ordenadas formando uma única lista ordenada para solucionar o problema de ordenação?

12) Explique sucintamente o funcionamento dos algoritmos quickSort, mergeSort e heapSort.

13) Qual a complexidade dos algoritmos quickSort, mergeSort e heapSort para o caso médio?

14) Quando um algoritmo, em uma passada de uma iteração toma o conjunto de dados e o divide em duas ou mais partes, tomando cada uma dessas partes e a processando separada e recursivamente e depois juntando os resultados, este algoritmo utiliza a técnica dividir-para-conquistar e será possivelmente:

- a)  $n \log n$
- b)  $\log n$
- c)  $n^2$
- d)  $n \log n^2$

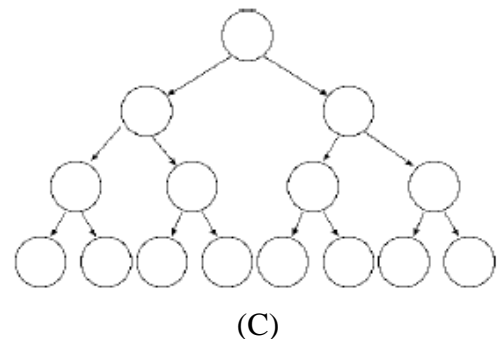
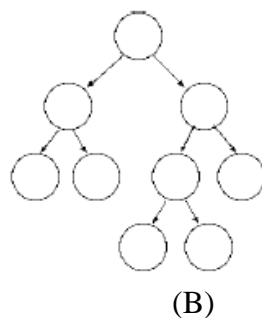
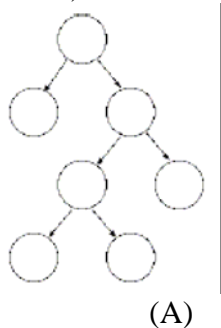
15) A pesquisa binária é um exemplo de:

- a)  $n \log n$
- b)  $\log n$
- c)  $n^2$
- d)  $n \log n^2$

16) Relacione a coluna da direita com a da esquerda:

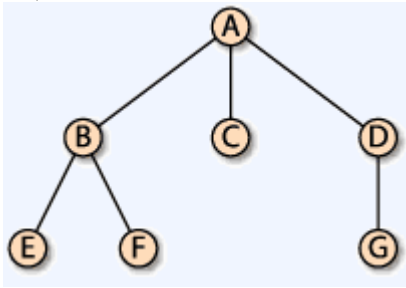
1.	Raiz	( )	Número de filhos de um nó.
2.	Caminho da árvore	( )	Conjunto finito de nós, que é vazio, ou consiste de um nó raiz e dois conjuntos disjuntos de nós, a subárvore esquerda e a subárvore direita.
3.	Grau	( )	Nós sem descendentes próprios.
4.	Altura de um nó	( )	Árvore em que se um nó tem alguma sub-árvore vazia então ele está no último nível.
5.	Nó	( )	Nó principal de uma árvore.
6.	Árvore Binária Cheia	( )	Elemento unitário da árvore. Deste elemento podem derivar ( <i>descender</i> ) outros elementos, designados de filhos, sendo o elemento atual o pai.
7.	Folhas	( )	Seqüência de nós consecutivos ( $n_1, n_2, \dots, n_{k-1}, n_k$ ) tal que existe sempre a relação $n_i$ é pai de $n_{i+1}$ .
8.	Árvore binária	( )	Número de nós do maior caminho de $n$ até um de seus descendentes

17) Relacione a letra da figura com o tipo da árvore:

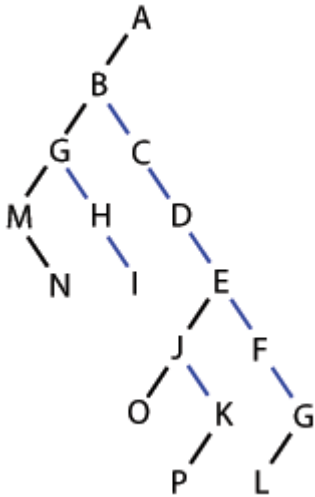


- ( ) Árvore binária completa
- ( ) Árvore binária cheia
- ( ) Árvore estritamente binária

18) Transforme a árvore abaixo em árvore binária:



19) Transforme a árvore binária abaixo em árvore genérica:

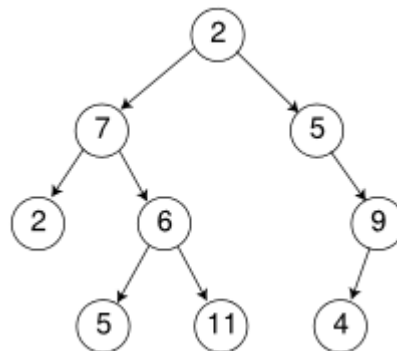
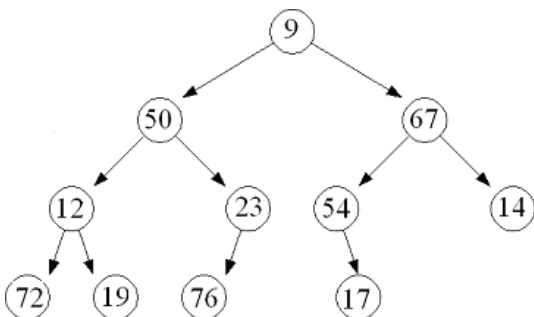


20) Verdadeiro ou falso?

( ) A idéia básica do algoritmo de ordenação bubble sort é montar uma árvore com os dados a serem ordenados, percorrer esses dados pela última camada denominada folhas e, a cada passagem, comparar cada elemento da folha com o seu sucessor. Se os elementos não estão ordenados deve-se trocá-los de posição.

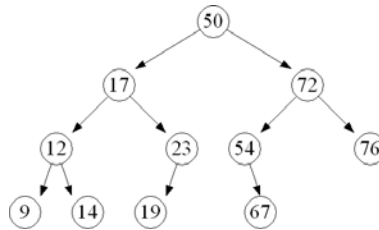
( ) Uma árvore binária é um conjunto finito de elementos que ou está vazio ou está dividido em 3 subconjuntos: um elemento chamado raiz da árvore e dois subconjuntos, cada um dos quais é, por si só, uma árvore binária, chamadas sub-árvore direita e sub-árvore esquerda.

21 – Determine os três tipos de percursos (InOrdem(Simétrica), PreOrdem e PosOrdem) das duas árvores binárias abaixo.



22 - Uma **Árvore de Busca Binária** (ou árvore de pesquisa binária) é uma árvore binária onde todos os nós são valores, todo nó a esquerda contém uma sub-árvore com os valores

menores ao nó raiz da sub-árvore e todos os nós da sub-árvore à direita contém somente valores maiores ao nó raiz. (Esta é a forma padrão, podendo ser invertida as sub-árvores dependendo da aplicação). Os valores são relevantes na árvore de busca binária. O objetivo desta árvore é estruturar os dados de forma flexível permitindo pesquisa binária. Vide exemplo abaixo.



Determine as Árvores de Busca Binária para as seguintes entradas abaixo:

- E, A, S, Y, Q, U, E, S, T, I, O, N (chaves da árvore são caracteres e não ints)
- 7 8 2 5 18 3 25 10 4 12 1 34 9 15 6

23) As duas operações básicas na ordenação simples são \_\_\_\_\_ itens e \_\_\_\_\_-los (ou algumas vezes \_\_\_\_\_-los).

24) Verdadeiro ou falso: (justifique se a resposta for falso)

( ) A ordenação pelo método da bolha sempre acaba comparando todo item com todos os outros.

25) Verdadeiro ou falso: (justifique se a resposta for falso)

( ) se houver N itens, a ordenação pelo método da bolha fará exatamente  $N^2$  comparações.

26) Na ordenação por seleção:

- as chaves maiores acumulam-se à esquerda (índices baixos)
- uma chave mínima é descoberta repetidamente;
- um número de itens tem que ser deslocados para inserir cada item em sua posição ordenada corretamente.
- os itens ordenados acumulam-se à direita.

25) Verdadeiro ou falso: (justifique se a resposta for falso)

( ) no quickSort, o pivô pode ser um elemento arbitrário do vetor.

26) Uma árvore binária será uma árvore de busca se:

- todo nó não folha tiver filhos cujos valores-chaves são menores (ou iguais) que o do pai.
- todo filho à esquerda tiver uma chave menor que a do pai e todo filho à direita tiver uma chave maior (ou igual) que a do pai.
- no caminho da raiz para cada nó folha, a chave de cada nó é maior (ou igual a) que a chave de seu pai.
- um nó pode ter um máximo de dois filhos.

27) No código Java para uma árvore, o \_\_\_\_\_ e a \_\_\_\_\_ são geralmente classes separadas.

28) Se uma árvore for representada por um vetor, o filho à direita de um nó com índice n terá um índice \_\_\_\_\_;

29) Em uma árvore binária completa com 20 nós e a raiz considerada como estando no nível 0, quantos nós existem no nível 4?