

Nome: _____

1. Marque a alternativa que representa, sucessivamente, os símbolos da análise assintótica de algoritmos: melhor, pior e médio.

(a) $O \ \Omega \ \Theta$
(b) $\Omega \ O \ \Theta$
(c) $\Omega \ \Theta \ O$
(d) $\Theta \ O \ \Omega$
(e) N.A

2. Determine a alternativa que representa a ordem decrescente de dominância assintótica:

(a) $O(1) \ O(\log n) \ O(n) \ O(n \log n) \ O(n^2) \ O(n^3) \ O(n^n) \ O(n!)$
(b) $O(n!) \ O(\log n) \ O(1) \ O(n \log n) \ O(n^3) \ O(n^2) \ O(n^n) \ O(1)$
(c) $O(n!) \ O(\log n) \ O(n) \ O(n \log n) \ O(n^n) \ O(n^3) \ O(n^2) \ O(1)$
(d) $O(n \log n) \ O(\log n) \ O(n) \ O(n!) \ O(n^n) \ O(n^2) \ O(n^3) \ O(1)$
(e) $O(n!) \ O(n^n) \ O(n^3) \ O(n^2) \ O(n \log n) \ O(n) \ O(\log n) \ O(1)$

3. Entre os tipos de algoritmos de ordenação a seguir, marque a alternativa que representa o merge-sort:

(a) Faz várias trocas sucessivas entre elementos até que todos estejam ordenados
(b) Calcula um valor médio e posiciona os valores maiores após o valor médio e os valores menores antes do valor médio, até que não seja possível fazer esta alocação
(c) A medida que um vetor é dividido por um valor médio, ocorre ordenação destes vetores resultantes, recursivamente, até que todo o vetor esteja ordenado
(d) A partir de um valor médio, um vetor é dividido sucessivamente, em uma primeira fase, e ordenado em uma segunda fase.
(e) N.A

4. (1.0) Em relação ao algoritmo a seguir, marque a alternativa que representa o valor do ponteiro b:

```
1  #include <iostream>
2  using namespace std;
3  int *a, *b, c;
4  int main()
5  {
6      a = new int;
7      b = new int;
8      *a = 5;
9      *b = 2;
10     c = 10;
11     b = a;
12     *b = ++(*a) + c;
13     cout << *b << endl;
14     return 0;
15 }
```

(a) 16 (b) 17 (c) 18 (d) 19 (e) N.A

5. Acerca do conteúdo geral de algoritmos avançados, avalie as questões a seguir e marque Certo (C) ou Errado (E):

a.() Uma função A possui uma dominância assintótica sobre uma função B. Pode-se concluir que o algoritmo que A representa possui menor desempenho.
b.() O Balanceamento em uma árvore AVL ocorre em intervalos de tempos determinado pelo algoritmo.

- c.() O nó raiz de uma árvore binária é o mesmo desde a inserção do primeiro valor
- d.() A análise assintótica determina o desempenho de um algoritmo a partir do tamanho dos dados de entrada
- e.() O algoritmo quicksort e o mergesort calculam um valor médio. O quicksort calcula a posição do meio de um vetor e o mergesort determina o valor do meio do vetor.

6. (1.0) Marque a alternativa que contém a característica que distingue o algoritmo de árvore AVL de uma árvore binária:

- (a) Uso de até três nós filhos para cada nó pai.
- (b) Realizar o balanceamento dos nós a partir do momento que a árvore estiver desbalanceada
- (c) Pré-alocação dos elementos maior que o valor médio para antes deste, permitindo a ordenação
- (d) Ordenação recursiva dos nós filhos até atingir os nós pais
- (e) N.A

7. Determine a complexidade assintótica do algoritmo a seguir para o pior caso:

```
1      int achaMaiorValor(int v[], int tamanho){
2          int maior = v[0];
3          for (int i=1; i < tamanho; i++) {
4              if (maior > v[i])
5                  maior = v[i];
6          }
7          return maior;
8      }
```

8. Determine a ordem assintótica para o algoritmo a seguir e marque a alternativa correta:

```
1      for (int i=0; i<tamanho; i++){
2          int k = 0;
3          for (int j=i+1; j<tamanho; j++){
4              k++;
5          }
6          cout << k << endl;
7      }
```

9. A partir dos dados a seguir: 9,4,10,5,3,13,15,8,7,2 crie uma árvore binária e imprima os valores na ordem in-fixada.

10. A partir dos dados a seguir: 12,2,4,7,15,18,12,26, 3 crie uma árvore binária e imprima os valores na ordem pós-fixada.

11. Escreva uma função recursiva que imprima somente o primeiro e o último valor de um vetor.

12. Escreva uma função recursiva que imprima a soma de todos os valores pares de um vetor.
