Algoritmos e Estruturas de Dados

Exame Final – 1ª Parte – V.2 12/JAN/2024

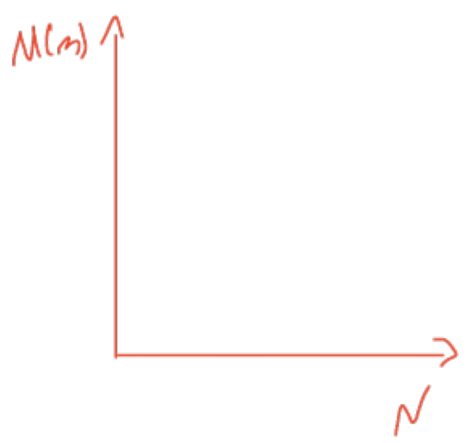
Assinale inequivocamente apenas a opção que considerar mais correta em cada pergunta.

1. Sejam dadas duas funções, t(n) ∈ g(n), uma constante positiva c e um inteiro não-negativo n0.
   a) Se t(n) ≤ c*g(n), para todo n > n0, então t(n) é da ordem de Ω(g(n)), i.e., t(n) pertence a Ω(g(n)).*
   b) Se t(n) ≥ cg(n), para todo n > n0, então t(n) é da ordem de O(g(n)), i.e., t(n) pertence a O(g(n)).
   c) Ambas estão corretas.
   d) Nenhuma está correta.

2. Considere (muitas) instâncias distintas de um mesmo problema, todas da mesma dimensão, e que foram resolvidas usando um mesmo algoritmo.
   a) Se o esforço computacional necessário para resolver cada uma dessas instâncias depende da configuração de cada instância, a análise da eficiência computacional do algoritmo implicará resultados distintos para o pior caso e para o caso médio.
   b) Se o esforço computacional necessário para resolver cada uma dessas instâncias foi idêntico, a análise da eficiência computacional do algoritmo implicará resultados distintos para o pior caso e para o caso médio.
   c) Ambas estão corretas.
   d) Nenhuma está correta.

3. A tabela abaixo apresenta o número de operações básicas efetuadas, por um determinado algoritmo, para sucessivos valores de n.

N	1	2	4	8	16	32	64	128	256	512	1024
M(n)	3	4	5	6	7	8	9	10	11	12	13



   Trata-se de um algoritmo com ordem de complexidade:
   a) quadrática.
   b) linear.
   c) logarítmica.
   d) Nenhuma está correta.

4. A tabela abaixo apresenta o número de operações básicas efetuadas, por um determinado algoritmo, para sucessivos valores de n.

N	1	2	4	8	16	32	64	128	256
M(n)	4	12	40	144	544	2112	8320	33024	1315584

   Trata-se de um algoritmo com ordem de complexidade:
   a) quadrática.
   b) linear.
   c) logarítmica.
   d) Nenhuma está correta.

5. A tabela abaixo apresenta o número de operações básicas efetuadas, por um determinado algoritmo, para sucessivos valores de n.

N	1	2	3	4	5	6	7	8	9	10
M(n)	1	2	5	3	8	13	21	34	55	89

a) Da tabela, obtém-se a seguinte relação recorrente para o número de operações, quando n >2: M(n) = M(n - 1) + M(n - 2).
b) Trata-se de um algoritmo com ordem de complexidade exponencial.
c) Ambas estão corretas.
d) Nenhuma está correta.

6. Considere o seguinte array de 6 elementos, que se pretende ordenar usando o algoritmo "Bubblesort".

0	1	2	3	4	5
6	5	4	3	2	1

a) São efetuadas 5 trocas entre elementos do array, para que seja ordenado por ordem crescente.
b) São efetuadas 15 trocas entre elementos do array, para que seja ordenado por ordem crescente.
c) São efetuadas 21 comparações entre elementos do array, para que seja ordenado por ordem crescente.
d) Nenhuma está correta.

7. Considere o seguinte array de 6 elementos, que se pretende ordenar usando a versão do algoritmo "Selectionsort" em que se começa por colocar o menor elemento na primeira posição.

0	1	2	3	4	5
6	5	4	3	2	1

a) São efetuadas 3 trocas entre elementos do array, para que seja ordenado por ordem crescente.
b) São efetuadas 15 comparações entre elementos do array, para que seja ordenado por ordem crescente.
c) Ambas estão corretas.
d) Nenhuma está correta.

8. Pretende-se ordenar um dado array de n elementos, todos distintos, e que se encontram armazenados de modo aleatório. A ordem de complexidade dessa tarefa depende do algoritmo escolhido, e será
a) $O(n^2)$, se for usado o algoritmo Bubblesort.
b) $O(n^2)$, se for usado o algoritmo de ordenação por seleção (Selectionsort).
c) O(n log n), se for usado o algoritmo Heapsort.
d) Todas estão corretas.

9. Pretende-se resolver o Problema das Torres de Hanói, para n discos.
a) Para n = 2 é necessário efetuar 4 movimentos de discos.
b) Para n = 3 é necessário efetuar 8 movimentos de discos.

c) O número de movimentos de discos efetuados é da ordem de O(2^n).
d) Todas estão corretas.

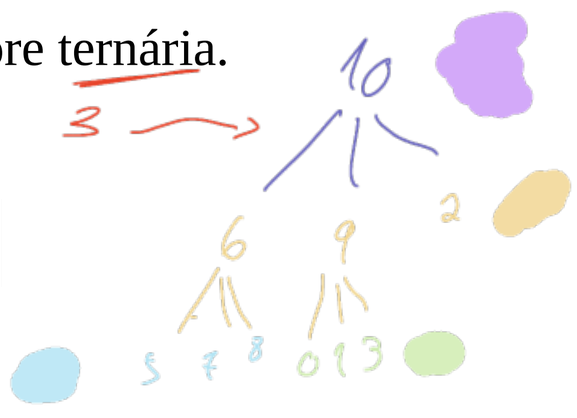
10. Seja dada uma escada com n degraus, que podem ser subidos um a um, dois a dois, ou três a três, ou numa qualquer combinação dos movimentos anteriores (ex., numa escada com três degraus, pode subir-se um só degrau e depois dois de uma só vez).
a) Para n = 3, é possível subir a escada apenas de 4 maneiras diferentes.
b) Para n = 4, é possível subir a escada apenas de 6 maneiras diferentes.
c) Ambas estão corretas.
d) Nenhuma está correta.

11. Seja dada uma árvore binária total, i.e., em que todos os níveis da árvore estão completamente preenchidos, com n nós.
a) O número de níveis da árvore é dado por $\lceil \log_2(n+1) \rceil$.
b) O número de nós que são folhas da árvore é dado por $\lfloor n \text{ div } 2 \rfloor$, em que div é o operador que determina o quociente da divisão inteira.
c) Ambas estão corretas.
d) Nenhuma está correta.

12. Seja dada uma árvore binária de altura equilibrada que armazena, de modo ordenado, n números inteiros.
a) No pior caso, concluir que um dado número não pertence à árvore é uma operação de complexidade O(log n).
b) No pior caso, determinar o valor do menor elemento armazenado na árvore é uma operação de complexidade O(log n).
c) Ambas estão corretas.
d) Nenhuma está correta.

13. O “array” abaixo armazena, por níveis, os elementos de uma árvore ternária.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
10	6	9	2	5	7	8	0	1	3	-	-	-	-	-



a) O elemento de valor 6 é filho do elemento de valor 9.
b) A árvore tem 7 folhas.
c) Ambas estão corretas.
d) Nenhuma está correta.

14. O array abaixo armazena, por níveis, os elementos de uma árvore binária de procura (“Binary Search Tree”). Atenção à ordem associada aos elementos da árvore:

0	1	2	3	4	5	6
8	4	12	2	6	10	14

a) se forem sucessivamente procurados cada um dos elementos do conjunto {2, 4, 6, 8, 10, 12, 14}, é efetuado um total de 17 consultas a elementos da árvore.
b) para procurar na árvore qualquer um dos elementos do conjunto {1, 3, 5, 7, 9, 11, 13}, são sempre consultados 3 elementos, concluindo-se depois que o valor procurado não pertence à árvore.

c) Ambas estão corretas.
d) Nenhuma está correta.

15. Considere um grafo orientado G(V,E), representado usando a lista ordenada dos seus nós e, para cada nó, a sua lista ordenada de adjacências.
a) No pior caso, adicionar uma nova aresta ao grafo orientado é uma operação de complexidade O(n).
b) No pior caso, verificar se um nó é isolado é uma operação de complexidade O(E).
c) Ambas estão corretas.
d) Nenhuma está correta.

16. A Teoria da Complexidade Computacional classifica diferentes problemas em classes de complexidade.
a) A classe P contém todos os problemas de decisão que podem ser resolvidos, em tempo polinomial, por um algoritmo determinista.
b) A classe NP contém todos os problemas de decisão cuja solução não pode ser verificada, em tempo polinomial, por um algoritmo determinista.
c) Ambas estão corretas.
d) Nenhuma está correta.

17. Em C++, a keyword auto permite:
a) inicializar, de modo automático, um variável com o seu default value.
b) deduzir, de modo automático, o tipo de uma variável em tempo de compilação.
c) Ambas estão corretas.
d) Nenhuma está correta.

18. Qual das seguintes estruturas de dados (C++ containers) fornece operações eficientes – isto é, com ordem de complexidade O(1) – para a inserção e remoção de elementos nas suas duas extremidades?
a) std::vector.
b) std::queue.
c) std::deque.
d) Nenhuma das anteriores.

19. Qual é a principal característica da estrutura de dados (C++ container) std::set?
a) Permite armazenar elementos duplicados.
b) Mantém os seus elementos ordenados.
c) Permite o acesso aleatório aos elementos armazenados.
d) Nenhuma está correta.

20. Em C++, o algoritmo std::count_if permite:
a) contar os elementos únicos de uma estrutura de dados (container).
b) contar o número de ocorrências de um elemento específico de uma estrutura de dados (container).
c) contar quantos elementos de uma estrutura de dados (container) satisfazem uma dada condição.
d) Nenhuma está correta.

Indique inequivocamente se cada uma das seguintes afirmações é Verdadeira ou Falsa.

F 1. Um algoritmo é definido por uma sequência de instruções possivelmente ambíguas, que permite resolver instâncias de um problema com tamanho finito em tempo finito.

V 2. Ao efetuar, usando o algoritmo clássico, o produto de matrizes A×B=C, em que a matriz A tem 10 linhas e 20 colunas e a matriz B tem 20 linhas e 10 colunas, são efetuadas 2000 multiplicações.

F 3. Um algoritmo que gera todas as permutações de um conjunto de n elementos é um algoritmo de ordem de complexidade exponencial, i.e., pertence a $O(2^n)$.

F 4. Uma função f(n) será de ordem de $\theta(g(n))$, i.e., f(n) pertence a $\theta(g(n))$, se existirem duas constantes positivas c e n_0 tais que $c \cdot g(n) \geq f(n)$, para todo $(n > n_0)$.

V 5. No caso médio, a versão iterativa do algoritmo de Pesquisa Binária num array ordenado tem ordem de complexidade O(log n).

F 6. Quando se usa a técnica da Programação Dinâmica, é necessário resolver repetidas vezes os mesmos sub-problemas. Verdadeiro ou Falso.

F 7. $1 + 2 + 4 + 8 + \ldots + 2^n - 1 = 2^n - 1$

V 8. $0 + 3 + 6 + 9 + \ldots + 3(n - 1) + 3n = 3(n^2+n)/2$

F 9. Quando se ordena um dado array de n elementos usando o algoritmo Heapsort, o primeiro passo é, habitualmente, transformar o array dado numa MIN-HEAP

V 10. Pretende-se ordenar um dado array de n elementos, todos distintos, e que se encontram armazenados de modo aleatório. Se for usado o algoritmos Heapsort, a ordem de complexidade dessa tarefa será O(n log n).

V 11. Numa árvore binária, o número máximo de nós do nível i é 2^i, considerando que a raiz da árvore pertence ao nível zero (i = 0).

F 12. Na travessia em Pós-Ordem de uma árvore binária, todos os elementos da subárvore direita da raiz são visitados primeiro que os elementos da subárvore esquerda da raiz.

V 13. Numa Árvore Binária de Procura (“Binary Search Tree”), a subárvore esquerda de um nó não pode conter elementos de valor superior a esse mesmo nó.

V 14. Uma árvore AVL é uma árvore binária equilibrada em altura em que, para cada nó, as alturas das suas duas subárvores diferem, sempre, de uma unidade.

V 15. Um grafo (não-orientado) completo, com n vértices, tem $(n^2 - n) / 2$ arestas.

V 16. Se um grafo orientado é fortemente conexo, não tem qualquer vértice isolado.

F 17. A travessia por níveis (“Breadth-First Traversal”) de um grafo orientado é habitualmente realizada usando uma pilha (“Stack”).

V 18. Considere um grafo orientado G(V,E), representado usando uma lista de vértices e, para cada vértice, a sua lista de adjacências. Se o grafo for completo, existem V listas de adjacências e o número total de nós definido às várias listas de adjacências é $V \times (V - 1)$.

19. Dado um grafo, um circuito Hamiltoniano é um caminho que partindo de um qualquer nó atravessa, uma única vez, cada um dos outros nós do grafo e regressa ao nó inicial.

20. Instâncias de grande dimensão do Problema do Caixeiro Viajante (“The Traveling Salesperson Problem”) não são habitualmente resolúveis em tempo útil.

Cotação:

Escolha múltipla: resposta certa 0,3 valores; resposta errada –0,1 valores.

Verdadeiro / Falso: resposta certa 0,2 valores; resposta errada –0,1 valores.