UNIRUY wyden

7064251680

07/10/2025 15:39

SM1 - A

Nome: Matrícula: _____

Disciplina: **ARA0174 / ALGORITMOS E COMPLEXIDADE** Data: __/__/____

Período: **2025.2 / SM1** Turma: **3001** NSQ: **13778290**

Leia com atenção as questões antes de responder.

É proibido o uso de equipamentos eletrônicos portáteis e consulta a materiais de qualquer natureza durante a realização da prova.

Boa prova.

1. _____ de 0,10

Marcar a alternativa correta sobre ponteiros:

A ☐ Ponteiros é um tipo de dado do tipo float que consegue armazenar outros tipos de dados.
B ☐ ponteiro é uma variável cujo conteúdo é um valor de variável.
C ☐ Todas as alternativas estão corretas
D ☒ ponteiro é uma variável cujo conteúdo é um endereço de memória
E ☐ Ponteiros é um tipo de dado do tipo int que consegue armazenar outros tipos de dados

2. _____ de 0,10

Qual é a complexidade de tempo do seguinte código em C?

```
#include

int exemplo1(int lista[]) {
  return lista[0];
}

int main() {
  int arr[] = {1, 2, 3, 4, 5};
  int resultado = exemplo1(arr);
  return 0;
}
```

A ☐ O(2^n)
B ☐ O(log n)
C ☐ O(n^2)
D ☐ O(n)
E ☒ O(1)

3. _____ de 0,10

Uma árvore binária T é uma estrutura de dados caracterizada ou por não ter elemento algum, ou por ter um elemento distinto, denominado raiz, com dois ponteiros para duas estruturas diferentes, denominadas subárvore esquerda e subárvore direita.

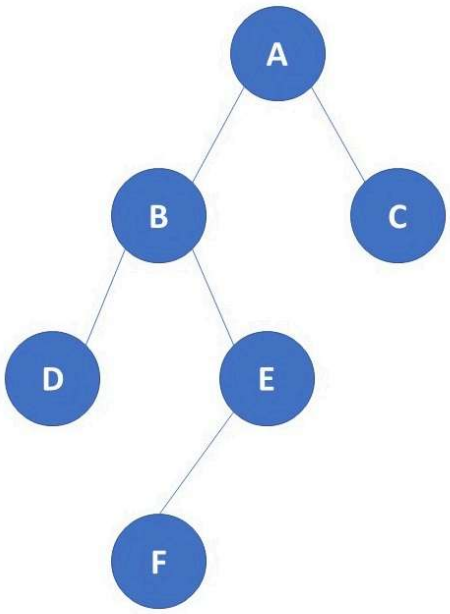

Figura01 – árvore binária

Assim, considerando árvore binária acima (figura 01), seguem as afirmações:

Nesse contexto podemos afirmar

I) A é um **nó raiz** de T e B é um **nó raiz** de uma subárvore de T;
II) B e C são **nós filhos** de A, assim como D e E são **nós filhos** de B;
III) Caso seja incluído um percurso direto entre D e F, a figura deixa de representar uma árvore e passa a ser um grafo;
IV) Os nós D, F, B e C são considerados **nós folha**.

Assim, assinale a alternativa correta:

A ☐ apenas as afirmações II, III e IV são corretas
B ☐ apenas as afirmações II e III são corretas
C ☐ todas as afirmações estão corretas
D ☒ apenas as afirmações I, II e III são corretas
E ☐ apenas as afirmações II e IV são corretas

4. _____ de **0,10**

Qual é a complexidade de tempo do seguinte código em C?

```
#include

void exemplo4(int lista[], int tamanho) {
  for (int i = 0; i < tamanho; i++) {
    for (int j = 0; j < tamanho; j++) {
      printf("%d %d\n", i, j);
    }
  }
}

int main() {
  int arr[] = {1, 2, 3, 4, 5};
  exemplo4(arr, 5);
  return 0;
}
```

A ☐ O(2^n)

B ☐ O(log n)
C ☒ O(n^2)
D ☐ O(n)
E ☐ O(1)

5. _____ de **0,10**

O uso de subrotinas (procedimentos e funções) permite que possamos criar programas modulares, deixando estes mais bem estruturado, o que nos leva ao desenvolvimento de programas com uma abordagem ¿Top-Down¿ ou ¿Botton-Up¿, onde vamos dividir o problema para conquistá-lo mais facilmente.

Diante do contexto apresentado podemos afirmar:

I) Geralmente criamos procedimentos e/ou funções para realizar tarefas que se repetem várias vezes na execução de um mesmo programa;
II) Um dos grandes benefícios de utilizarmos funções e/ou procedimentos é não precisar copiar o código todas as vezes que precisar executar aquela operação, além de deixar a leitura do código mais intuitiva e facilitar a manutenção do programa;
III) Os procedimentos e as funções, quando são chamados pelo programa principal, não possuem acesso as variáveis globais do sistema.
IV) Ao definirmos uma variável local, dentro de uma função ou procedimento, precisamos ter cuidado para não atribuirmos a ela o mesmo nome de uma variável global, pois essa atribuição não é permitida.

Assinale a alternativa correta:

A ☒ apenas as afirmações I e II são verdadeiras;
B ☐ apenas as afirmações II, III e IV são verdadeiras;
C ☐ todas as afirmações são verdadeiras;
D ☐ apenas as afirmações I, II e III são verdadeiras;
E ☐ apenas as afirmações I, II e IV são verdadeiras;

6. _____ de **0,10**

A recursividade se apresenta como uma ferramenta poderosa para resolução de problemas computacionais, tanto acadêmicos quanto comerciais. Por isso, vários algoritmos de ordenação e estrutura de dados são baseados em programas recursivos.

Sobre recursividade podemos afirmar:

I) - É importante saber quando utilizar a **recursão** ou a **iteração** na solução de um problema. Em linhar gerais, solucionar problemas através da **iteração** geralmente exige uma quantidade maior de linhas de código;

II) - Quando o foco é a eficiência, a **iteração** é mais rápida em todos os casos;

III) - Em algumas situações as **funções recursivas** podem não oferecer benefícios quando comparadas a um algoritmo elaborado utilizando **iteração**;

IV) - Quando um **algoritmo recursivo** faz muitas chamadas, ele pode comprometer o funcionamento adequado da solução ao causar uma sobrecarga no uso de memória;

Diante das afirmações acima, assinale a alternativa correta:

A ☐ apenas as afirmações II, III e IV estão corretas;
B ☐ todas as afirmações estão corretas;
C ☐ apenas as afirmações I e III estão corretas;
D ☒ apenas as afirmações I, III e IV estão corretas;
E ☐ apenas as afirmações I, II e III estão corretas;

7. _____ de **0,10**

A notação Big O é uma ferramenta importantíssima para os cientistas da computação analisarem o custo de um algoritmo.

Tecnocamente falando, a notação Big O é uma notação matemática que descreve o comportamento limitante de uma função quando o argumento tende a um valor específico ou ao infinito. Ela pertence a uma família de notações inventadas por Paul

Bachmann, Edmund Landau e outros, coletivamente chamadas de notação Bachmann¿Landau ou de notação assintótica. Ou seja, a notação Big O descreve a complexidade do seu código usando termos algébricos.

Diante do contexdo apresentado, podemos afirmar:

I) Quando tentamos descobrir a Big O para uma função g(n) específica, nos preocupamos apenas com o termo dominante da função. O termo dominante é o termo que cresce mais rápido.
II) Na notação Big O, podemos dizer que um algoritmo tem a menor complexidade quando seu custo for O(1).
III) De uma forma geral, a complexidade de um algoritmo tem relação principalmente com seu tempo de execução e espaço de memória ocupado para execução;
IV) Na notação Big O, podemos dizer que um algoritmo que tem a complexidade O(log(n)) é mais complexo do que que custa $O(n^4)$;

Assinale a alternativa correta:

A ☐ apenas as afirmações I e IV são verdadeiras;
B ☐ apenas as afirmações II, III e IV são verdadeiras;
C ☐ todas as afirmações são verdadeiras;
D ☒ apenas as afirmações I, II e III são verdadeiras;
E ☐ apenas as afirmações I, II e IV são verdadeiras;

8. _____ de 0,10

Considerando o seguinte conjunto de nós (V - vértices) e arestas (E - arestas) de um grafo G:

- V = {A, B, C, D, E}
- E = {(A,B), (B,C), (C,E), (E,D), (D,A)}

A representação gráfica (desenho) do grafo acima resulta em qual classe de grafo?

A ☐ Grafo Árvore.
B ☐ Grafo 3-regular.
C ☐ Grafo nulo.
D ☐ Grafo completo.
E ☒ Grafo ciclo.

9. _____ de 0,10

Considere as seguintes afirmações sobre a complexidade computacional das Árvores Binárias de Busca (BST):

- I. No melhor caso, a árvore binária pode estar balanceada, levando o processo de busca, inserção e remoção ao custo computacional `O(n log n)`.
- II. No pior caso, a árvore binária pode estar completamente desbalanceada, levando o processo de busca, inserção e remoção ao custo computacional de ordem linear, ou seja, `O(n)`.
- III. O custo computacional de inserir um nó em uma árvore binária no melhor cenário é da ordem logaritmica, ou seja, `O(log n)`.

Estão **corretas** as afirmativas:

A ☐ I e III.
B ☒ II e III.
C ☐ Somente a III.
D ☐ I, II e III.
E ☐ I e II.

10. _____ de 0,10

Utilizamos a notação *O* para classificar o desempenho de um algoritmo. Dizemos que f=O(g) quando para qualquer valor de n maior que n0 a função cg(n)>f(n).
A análise da complexidade computacional de um algoritmo é sempre feita na visão do pessimista, isto é, na

análise do pior caso apresentado ao algoritmo.
O método da bolha é um algoritmo de ordenação baseado em trocas. A análise de complexidade de pior caso do método da bolha diz que a complexidade computacional do algoritmo é On2, claro, para o pior caso. . É conhecido que nos melhores casos, o método da bolha executa em um tempo proporcional à On, porém, não levamos isto em consideração para análise do complexidade com a visão do pessimista. Um outro método de ordenação é o merge sort, sua complexidade computacional é On logn para todas as instâncias, ou seja, todos as instâncias submetidas ao merge sort são do pior caso. Com base nestes fatos, interprete as afirmações abaixo segundo seu valor verdade.

I - Pode-se afirmar que o merge sort SEMPRE executa mais rápido que o método da bolha, para a mesma instância submetida aos dois algoritmos.
II - Pode-se afirmar que o merge sort executa mais rápido que o método da bolha quando instâncias do pior caso do método da bolha são submetidas a ambos os algoritmos.
III - Existe a possibilidade de existirem instâncias que executam mais rápido no método da bolha do que no merge sort. Entretanto, estas instâncias não são levadas em consideração porque não devem fazer parte da análise do pessimista.

A ☐ Somente III é verdadeira.
B ☐ Somente I é verdadeira
C ☐ Somente II é verdadeira.
D ☒ Somente II e III são verdadeiras.
E ☐ Somente I e II são verdadeiras.