



---

## **Edital de Seleção 032/2016 PROPESP/UFAM**

### **Prova de Conhecimento**

### **Caderno de Questões**

---

#### **CANDIDATO:**

INSCRIÇÃO:

\_\_\_\_\_

Assinatura conforme identidade

---

#### **INSTRUÇÕES PARA O CANDIDATO:**

- Verifique o seu nome e o número da sua inscrição impressos neste CADERNO DE QUESTÕES. Assine seu nome no local apropriado somente quando autorizado pelo aplicador da prova, no momento da identificação.
- As respostas a todas questões devem ser preenchidas na FOLHA DE RESPOSTAS, no campo correspondente a cada questão.
- Em nenhuma hipótese haverá substituição deste CADERNO DE QUESTÕES por erro de preenchimento do candidato.
- Este CADERNO DE QUESTÕES ficará disponível aos candidatos a partir do dia 22/02/2016, na site do PPGI.

### QUESTÃO 01

Suponha que você recebeu a tarefa de depurar uma implementação de Quicksort, cujo objetivo é para classificar um vetor em ordem crescente. Considere que o pivô escolhido terá o valor 4 e que você executou até o momento apenas o particionamento completo gerado com o pivô, sem ter realizado ainda qualquer chamada recursiva. Sobre as afirmações:

4	1	2	3	17	2	22	3
---	---	---	---	----	---	----	---

- I) Dependendo da implementação da partição, a ordem dos elementos nesse momento pode ser 3, 1, 2, 3, 2, 17, 22 e 4
- II) É impossível que uma implementação de Quicksort tenha escolhido o 4 como pivô
- III) Dependendo da implementação da partição, a ordem dos elementos nesse momento poderia ser 3, 1, 2, 3, 2, 4, 17 e 22
- IV) Dependendo da implementação da partição, a ordem dos elementos nesse momento poderia ser 4, 1, 2, 3, 2, 3, 17 e 22
- a) Apenas I está correta  
b) Apenas II está correta  
c) Apenas I e III estão corretas  
d) Apenas III e IV estão corretas  
e) Apenas III está correta

### QUESTÃO 02

Quantas vezes o símbolo '#' é impresso pela função teste quando chamada com parâmetro 5 (**teste(5)**)?

```
void teste (int i) {  
    if (i > 1) {  
        teste (i/2);  
        teste (i/2);  
    }  
    printf( "#"); /*imprime símbolo # na tela */  
}
```

- a) 3  
b) 4  
c) 7  
d) 8  
e) Nenhuma das alternativas anteriores

### QUESTÃO 03

Quando um novo nó é removido no meio de uma lista encadeada dinâmica, qual das seguintes afirmações é verdadeira:

- a) Somente os nós que aparecem após o novo nó precisam ser movidos  
b) Somente os nós que aparecem antes do novo nó precisam ser movidos  
c) Os nós que aparecem antes do novo nó e o último nó precisam ser movidos  
d) Os nós que aparecem depois do novo nó e o primeiro nó precisam ser movidos  
e) Nenhuma das opções acima



#### QUESTÃO 04

Se você tem que implementar um algoritmo de ordenação que tem um tempo de execução competitivo quando comparado a outros, e que não varia muito seu desempenho em função da entrada de dados fornecida. Qual das opções é melhor?

- a) Ordenação por inserção
- b) Heapsort
- c) Quicksort
- d) Tanto Quicksort quanto Heapsort atendem aos requisitos
- e) Hash

#### QUESTÃO 05

Se em uma árvore binária de pesquisa a diferença de altura entre a sub-árvore a esquerda e a sub-árvore a direita de um dos nós for maior que 1, pode-se concluir que:

- a) A árvore é uma árvore AVL
- b) Trata-se com certeza de uma árvore binária sem balanceamento
- c) Trata-se com certeza de uma árvore vermelho e preto
- d) Tal árvore não pode existir por não se tratar de uma árvore binária de pesquisa
- e) Nenhuma das alternativas anteriores

#### QUESTÃO 06

Uma estrutura de dados bastante útil quando deseja-se armazenar elementos para recupera-los posteriormente em ordem reversa a de inserção:

- a) A árvore é uma árvore AVL
- b) Um hash
- c) Uma fila de prioridades com ordenação reversa de prioridade
- d) Uma pilha
- e) Uma árvore vermelho e preto

#### QUESTÃO 07

Considere as afirmações sobre busca binária:

- i) é utilizada apenas sobre vetores ordenados
- ii) pode determinar se uma chave está ou não com um número de comparações de chave inferior a  $2\log_2 n$  passos, onde  $n$  é o tamanho do vetor.
- iii) Supera a busca sequencial quando o vetor é grande, por exemplo com mais que mil elementos, e está ordenado.

- a) Apenas as afirmações II e III estão corretas
- b) Apenas as afirmações I e II estão corretas
- c) Apenas as afirmações I e III estão corretas
- d) Todas as afirmações estão corretas
- e) Nenhuma das alternativas anteriores

### QUESTÃO 08

Considere os códigos de funções abaixo, assumindo que os mesmos são usados dentro de um programa e que os todos os cabeçalhos necessários são incluídos:

<pre>/*QUADRO I) Código para inserir em uma lista encadeada. Cada nó da lista tem um campo <b>prox</b>, que é usado para encadea-lo ao próximo elemento, e tem um <b>dado</b> do tipo int. */  /* prim é um ponteiro para o ponteiro ao primeiro elemento da lista */  void insere(tipoNo **prim, int d) {     tipoNo *aux;     aux = (tipoNo *)malloc(sizeof(tipoNo));     if(aux) { aux -&gt;prox = *prim;               *prim = aux;               aux -&gt; dado = d;             } }</pre>	<pre>/*QUADRO II) Código para remover primeiro elemento de uma lista encadeada. Cada nó da lista tem um campo <b>prox</b>, que é usado para encadea-lo ao próximo elemento, e tem um <b>dado</b> do tipo int. */  /* prim é um ponteiro para o ponteiro ao primeiro elemento da lista */  1. void remove (tipoNo **prim){ 2. tipoNo *aux; 3. if(*prim) { aux = *prim; 4.     *prim = aux-&gt;prox; 5.     free(*prim); 6.     *prim = aux; 7. } 8. }</pre>
---	--

Marque a alternativa correta dentre as afirmações abaixo:

- Os dois códigos executam as tarefas propostas sem gerar qualquer tipo de erro de lógica, de vazamento de memória ou falha de segmentação
- Apenas os código I realiza sem erros a tarefa descrita nos comentários
- Apenas os código II realiza sem erros a tarefa descrita nos comentários
- Os dois códigos não executam suas tarefas como descrito nos comentários, mas também não geram vazamento de memória
- O primeiro código gera vazamento de memória e o segundo não realiza a tarefa descrita nos comentários

### QUESTÃO 09

Considere as afirmações sobre um hash por encadeamento:

- Suas colisões secundárias podem prejudicar seu desempenho
- Não funciona com número de chaves maior que o tamanho da tabela hash
- O número médio de operações esperado para realizar uma inserção é constante quando computado em função do tamanho da tabela se o hash for bem projetado

Sobre as afirmações, sabe-se que:

- Todas são verdadeiras
- Todas são falsas
- Apenas I e III são verdadeiras
- Apenas III é verdadeira
- Apenas I e II são verdadeiras

### QUESTÃO 10

Considere os códigos de funções abaixo, assumindo que os mesmos são usados dentro de um programa e que os todos os cabeçalhos necessários são incluídos:

<pre>/*QUADRO I) função que retorna 1 se a chave de busca <b>chave</b>, passada como parâmetro, ocorre dentro de um vetor de inteiros ordenado e retorna 0 caso a chave não ocorra no vetor. O vetor tem tamanho determinado pelo parâmetro tm */  int buscaChave(int v[], int chave, int tm) {  while((tm &gt;= 0) &amp;&amp; (v[tm] != chave)) tm --; if (tm &gt;= 0) return 1; return 0; }</pre>	<pre>/* QUADRO II) Conta ocorrências de uma chave na lista encadeada, retornando o total de ocorrências , zero caso a chave não ocorra */ int ContaOcorrenciasChave(tipoNo *prim, int chave) { tipoNo *aux = prim; int cont = 0; int x; for (x = 0; aux[x] != NULL; x++){ if(aux[x]. dado == chave) cont++; } return cont; }</pre>
---	--

Marque a alternativa correta dentre as afirmações abaixo:

- Os dois códigos executam as tarefas propostas sem gerar qualquer tipo de erro de lógica ou de vazamento de memória
- Apenas os código I executa sem erros a tarefa proposta
- Apenas o código II executa sem erros a tarefa proposta
- Os dois códigos geram erros de vazamento de memória ou de falha de segmentação
- Nenhuma das alternativas anteriores

### QUESTÃO 11

Considere as afirmações sobre um hash por encadeamento:

- Tem como desvantagem o fato de não ordenar as chaves inseridas
- O número médio de operações esperado para realizar uma busca é constante quando computado em função do tamanho da tabela se o hash for bem projetado
- Mesmo quando bem implementado, ocupa mais espaço extra do que uma árvore-b em memória.

Sobre as afirmações, sabe-se que:

- Todas são verdadeiras
- Todas são falsas
- Apenas I e III são verdadeiras
- Apenas III é verdadeira
- Apenas I e II são verdadeiras

### QUESTÃO 12

Marque a resposta correta. Ao inserirmos as chaves 1, 2 e 3, 4, 5, 6 em uma árvore AVL, sua altura (medida como a quantidade máxima de nós em um caminho entre a raiz e uma folha da árvore, incluindo a raiz e a folha na conta) será:

- a) 2
- b) 3
- c) 4
- d) 5
- e) 6

### QUESTÃO 13

Sobre uma árvore binária de pesquisa sem balanceamento, pode-se afirmar que:

- I) São estruturas de dados que, quando degeneradas, têm custo de inserção proporcional ao número de chaves inseridas
- II) Não servem para armazenar dados quando deseja-se ter acesso aos mesmos de maneira ordenada
- III) Têm como principal custo extra de memória os dois apontadores que indicam os nós à esquerda e à direita de cada nó criado

Sobre as afirmações, sabe-se que:

- a) Apenas I é verdadeira
- b) Apenas I e II são verdadeiras
- c) Todas as afirmações são verdadeiras
- d) Apenas II e III são verdadeiras
- e) Nenhuma das alternativas anteriores está correta

### QUESTÃO 14

Sobre árvores binárias de pesquisa sem balanceamento, pode-se afirmar que:

- I) É uma estrutura de dados inerentemente recursiva, sendo impossível implementar uma busca em uma árvore de maneira não recursiva
- II) É uma estrutura de dados inerentemente recursiva, sendo impossível implementar uma inserção em uma árvore de maneira não recursiva
- III) Em muitos casos práticos, pode ter custo de busca com número de operações proporcional a  $\log n$ , onde  $n$  é o número de elementos inseridos na árvore

Sobre as afirmações, sabe-se que:

- a) Apenas I e II estão corretas
- b) Apenas III está correta
- c) Apenas I e III estão corretas
- d) Apenas II e III estão corretas
- e) Nenhuma das alternativas anteriores



### QUESTÃO 15

Marque a resposta correta. Ao inserirmos as chaves 1, 2 e 3, 4, 5, 6 em uma árvore AVL, a diferença de altura entre as sub-árvores à esquerda e à direita do nó raiz será:

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

### QUESTÃO 16

Considere um vetor não ordenado utilizado para armazenar chaves únicas (sem repetição). Pode-se afirmar que:

- I) O melhor algoritmo para busca no vetor é a busca binária mesmo que o vetor não esteja ordenado, independente do número de elementos do vetor e da quantidade de buscas a serem realizadas
- II) Uma busca binária implementada no vetor não funcionaria adequadamente, fornecendo resultados errados a menos que fosse feita uma ordenação
- III) Um algoritmo de busca sequencial funcionaria e daria resultados comparando a chave com todas as posições do vetor na pior das hipóteses

Sobre as afirmações, sabe-se que:

- a) Apenas as afirmações I e III estão corretas
- b) Apenas as afirmações II e III estão corretas
- c) Todas as afirmações estão corretas
- d) Todas as afirmações estão erradas
- e) Nenhuma das alternativas anteriores

### QUESTÃO 17

Considere as afirmações abaixo:

- I) O RADIX sort não realiza comparação entre chaves para realizar a ordenação de um vetor
- II) O número de vezes que é necessário passar por todos os elementos do vetor no RADIX sort é proporcional ao número médio de dígitos dos elementos do vetor na base utilizada pelo algoritmo para ordenação
- III) O RADIX sort requer um espaço extra de armazenamento proporcional ao tamanho do vetor

Sobre as afirmações, sabe-se que:

- a) Apenas a afirmação I é verdadeira
- b) Apenas a afirmação II é verdadeira
- c) Apenas a afirmação III é verdadeira
- d) Apenas as afirmações II e III são verdadeiras
- e) Nenhuma das alternativas anteriores

### QUESTÃO 18

Considere que o vetor abaixo será ordenado com o algoritmo de ordenação por seleção

4	5	2	3	17	2	1	22
---	---	---	---	----	---	---	----

Considerando-se que o vetor tem sua primeira posição contada como 0 (zero), a primeira troca de valores entre posições do vetor ocorrerá entre:

- a) posição 0 e posição 2
- b) posição 0 e posição 7
- c) Posição 0 e posição 6
- d) Posição 1 e posição 6
- e) Posição 3 e posição 5

### QUESTÃO 19

Considere os vetores abaixo

VetI

1	2	3	4	5	6	8	7
---	---	---	---	---	---	---	---

VetII

7	8	8	5	4	3	2	1
---	---	---	---	---	---	---	---

E considere as afirmações a seguir:

- I) A ordenação por inserção é melhor que a ordenação por seleção em VetI se considerarmos o número de comparações entre chaves realizadas pelos dois algoritmos
- II) O RADIX sort executado em base 10 realiza um número menor de operações (instruções) nos dois casos quando comparado ao mergesort
- III) O RADIX sort executado em base 10 realiza um número menor de operações apenas em VetII quando comparado ao mergesort, ocorrendo o oposto em VetI

Sobre as afirmações, sabe-se que:

- a) Apenas a afirmação I é verdadeira
- b) Apenas a afirmação II é verdadeira
- c) Apenas a afirmação III é verdadeira
- d) Apenas as afirmações I e II são verdadeiras
- e) Apenas as afirmações I e III são verdadeiras





---

### QUESTÃO 20

Considere as afirmações abaixo:

- I) Pode-se determinar o maior elemento de um vetor ordenado a um custo constante (realizando-se um número constante de operações)
- II) Pode-se inverter a ordem de um vetor já ordenado fazendo-se um número de trocas de posições igual à metade do tamanho do tamanho do vetor e sem ter que criar uma cópia do mesmo
- III) O Quicksort é sempre a opção mais rápida (com menor tempo de execução) para ordenar qualquer vetor de qualquer tamanho e com qualquer ordenação inicial das chaves

Sobre as afirmações, sabe-se que:

- a) Todas as afirmações são verdadeiras
- b) Apenas I é verdadeira
- c) Apenas I e III são verdadeiras
- d) Apenas I e II são verdadeiras
- e) Nenhuma das alternativas anteriores