



Edital de Seleção 042/2015 PROPESP/UFAM

Prova de Conhecimento para Candidatos ao Mestrado

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 13/02/2016, na Secretaria do PPGI, ou poderá ser solicitado via e-mail a secretaria do PPGI.
- **As questões 1, 2, 4, 5 e 6 valem 1,0 ponto cada. As outras questões valem 0,5 ponto cada uma.**

ALGORITMOS E ESTRUTURA DE DADOS

QUESTÃO 01. Considere o vetor abaixo com posições que vão da posição zero até a posição dez. Qual das alternativas abaixo representa a sequência exata de valores a serem comparados com a chave 102 em uma busca binária.

4	6	10	13	20	38	40	45	50	60	100
---	---	----	----	----	----	----	----	----	----	-----

- a) 6; 13; 38; 45; 60
- b) 38; 50; 60; 100
- c) 38; 45; 60
- d) 4; 6; 10; 13; 20; 38; 40; 45; 50; 60; 100
- e) 6; 13; 38; 45; 60; 100

QUESTÃO 02. Inseriu-se em uma pilha os valores A,B,C e D, seguindo essa ordem. Se logo após são executadas duas operações de remoção, pode-se dizer que:

- a) Os valores removidos serão A e B, nessa ordem
- b) Os valores removidos serão C e D, nessa ordem
- c) Os valores removidos serão D e C, nessa ordem
- d) A resposta depende da chave de busca, pois a remoção depende da chave fornecida
- e) Nenhuma das alternativas anteriores

QUESTÃO 03. Complete com a opção correta: _____ é uma estrutura de dados normalmente utilizada em aplicações onde há vários consumidores compartilhando um recurso que é utilizado continuamente, com o servidor dando acesso a um consumidor a cada instante do tempo

- a) pilha de compartilhamento
- b) fila encadeada dinâmica
- c) lista circular
- d) função de compartilhamento binária
- e) busca binária

QUESTÃO 04. Você precisa ordenar um arquivo com registros de tamanho fixo cuja chave é um inteiro. Considerando que:

- I. Há espaço em memória principal para ordenar o arquivo, mas não há espaço para duas cópias do arquivo na memória;
- II. O arquivo não contém chaves repetidas;
- III. O arquivo está com seus registros praticamente ordenados, com apenas um número pequeno de itens, inferior a 3, fora do lugar, mas você não sabe quais são tais itens;
- IV. A chave de ordenação é uma cadeia de caracteres (string) de tamanho longo, superior a 200 bytes;
- V. O arquivo possui milhões de itens.

Qual a melhor opção de algoritmo de ordenação em termos de tempo de execução, considerando também a viabilidade, dentre as alternativas abaixo ?

- a) Quicksort
- b) Mergesort
- c) Inserção
- d) Radixsort
- e) Busca binária

QUESTÃO 05. Considere os códigos de funções abaixo:

<pre>/*QUADRO I) Código para inserir em uma lista encadeada. Cada nó da lista tem um campo prox, que é usado para encadea-lo ao próximo elemento, e tem um dado 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)); aux -> prox = *prim; *prim = aux; aux -> dado = d; }</pre>	<pre>/*QUADRO II) Código para remover primeiro elemento de uma lista encadeada. Cada nó da lista tem um campo prox, que é usado para encadea-lo ao próximo elemento, e tem um dado do tipo int. */ void remove (tipoNo **prim){ tipoNo *aux; aux -> prox = *prim; *prim = aux free(*prim); }</pre>
<pre>/*QUADRO III) função que retorna 1 se a chave de busca chave, passada como parâmetro, ocorre dentro de uma lista encadeada pelo menos uma vez e retorna 0 caso a chave não ocorra na lista */ int buscaChave(tipoNo *prim, int chave) { tipoNo *aux = prim; int retorno = 0; while(aux != NULL) { if(aux->dado == chave) { retorno = 1; } else { retorno = 0; } aux = aux-> prox; } return retorno; }</pre>	<pre>/* QUADRO IV) 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:

- Apenas os códigos dos quadros I e IV estão corretos
- Apenas os códigos dos quadros II e III estão corretos
- Apenas os códigos dos quadros I e III estão corretos
- Apenas os códigos dos quadros II e IV estão corretos
- Nenhuma das alternativas anteriores

QUESTÃO 06. Quando um vetor é ordenado, podem ocorrer casos em que alguns valores começam na mesma posição em que vão terminar. Por exemplo, no vetor que é originalmente (50 0 20 4 60 45 90) o 20 está inicialmente exatamente na posição em que deve ficar no final, terceira posição, quando o vetor estiver ordenado (0 4 20 45 50 60 90).

Entretanto, durante o processo de ordenação, alguns algoritmos podem (dependendo do algoritmo) mover o elemento de lugar antes de chegar ao final da ordenação (claro que no fim das contas todo algoritmo correto acabará colocando o elemento de volta ao lugar correto). Considere o caso genérico de um elemento estar na posição inicial e nunca ser movido, o exemplo acima serve apenas de ilustração.



Qual(quais) das afirmações é(são) correta(s)?

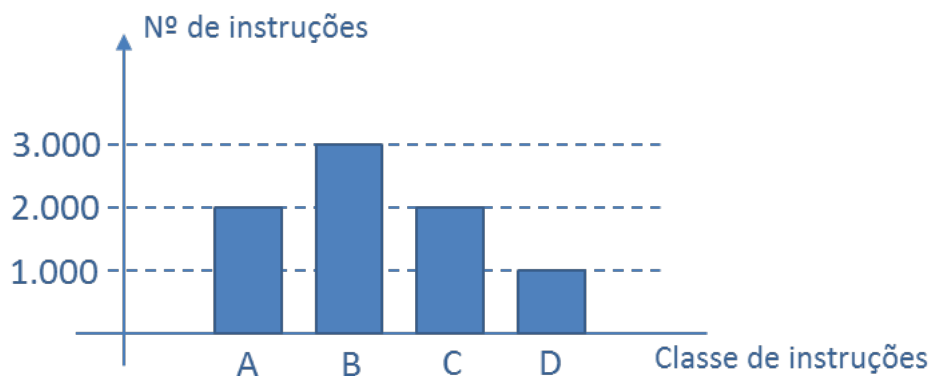
- I. Seleção nunca (mesmo temporariamente) moveria tal tipo de elemento para outra posição do vetor.
- II. Inserção nunca (mesmo temporariamente) moveria tal tipo de elemento para outra posição do vetor.
- III. Mergesort nunca (mesmo temporariamente) moveria tal tipo de elemento para outra posição do vetor.

- a) Somente I
- b) Somente II
- c) Somente III
- d) Somente I e II
- e) Nenhuma das alternativas anteriores

ARQUITETURA DE COMPUTADORES

QUESTÃO 07. Um programa é executado em uma arquitetura de computador dotada de apenas um nível de memória cache. A taxa de falha (*miss rate*) dessa memória cache é de 1% e a penalidade de falha (*miss penalty*) é de 10ns.

As instruções desse programa estão distribuídas em um total de quatro classes, conforme gráfico abaixo. Delas, apenas as instruções da classe C implicam em acesso à memória de dados. O tempo de execução total do programa, considerando o tempo para tratamento de falhas da cache, é de 260ns.



Para reduzir o tempo de execução do programa, o fabricante resolve inserir uma cache de nível 2, mantendo os demais parâmetros de projeto inalterados. Esta alteração provocou uma redução da taxa de falha resultante (considerando os dois níveis de cache em conjunto) para 0,5%, mas elevou a penalidade de falha para 15ns.

Nessa nova configuração, o tempo de execução desse mesmo programa será de:

- a) 150ns
- b) 160ns
- c) 200ns
- d) 210ns
- e) 250ns

QUESTÃO 08. Avalie as seguintes afirmações, sobre memória virtual:

I – Para páginas de 4 KB de tamanho, o 12 bits mais significativos do endereço virtual são usados para identificar uma página, e os 20 bits menos significativos são usados como deslocamento interno da página.

II – A falta de uma página na tabela de página (*page fault*) implica em uma falha na cache (*miss*) dos dados contidos nessa página.

III – O posicionamento de páginas da memória virtual em quadros da memória principal tipicamente utiliza a política de mapeamento direto.

IV – A política de substituição LRU (*Least Recently Used*) explora o princípio de localidade espacial.

Está correto o que se afirma em:

- a) II, apenas.
- b) I e II, apenas.
- c) I e IV, apenas.
- d) I, II e IV, apenas.
- e) I, II, III e IV.

QUESTÃO 09. Abaixo, são exibidos quatro trechos de código assembly, identificados pelas letras A, B, C, D. Essa linguagem utiliza as seguintes convenções:

- A instrução **lw** (load word) carrega valores armazenados em memória para um registrador indicado;
- A instrução **sw** (store word) faz o inverso;
- A instrução **beq** (branch on equal) desvia a execução de um programa para uma instrução indicada por um rótulo, se dois registradores indicados forem iguais.

A	B	C	D
lw R4, 0(R0) add R5, R2, R4	lw R3, 0(R0) add R3, R2, R1	sw R2, 0(R0) add R2, R1, R3	beq R6, R3, LABEL add R0, R1, R6

Cada trecho de código apresenta uma dependência, o que pode vir a causar um hazard na execução em pipeline. Verifique qual dependência que cada trecho apresenta, associando um número de 1 a 4 conforme a seguinte convenção:

- (1) Dependência de dados (hazard RAW)
- (2) Antidependência (hazard WAR)
- (3) Dependência de saída (hazard WAW)
- (4) Dependência de controle (hazard de controle)

A associação correta é:

- a) A-1, B-4, C-2, D-3
- b) A-1, B-3, C-2, D-4
- c) A-2, B-3, C-1, D-4
- d) A-2, B-4, C-1, D-3
- e) A-3, B-1, C-2, D-4

FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO

QUESTÃO 10. Dada a linguagem $L = \{0^m 1^n \mid m \neq n, m \geq 1, n \geq 1\} \cup \{0^p 1^p \mid p \geq 1\}$, analise as seguintes afirmações:

- I. Ou existe um autômato finito não-determinístico que reconhece L, ou existe um autômato finito determinístico que reconhece L, mas não ambos;
- II. A linguagem L pode ser representada pela expressão regular $00^*(1^*)^*1$;
- III. A linguagem L pode ser representada pela expressão regular 0^*011^* ;
- IV. A linguagem L pode ser gerada pela gramática $G = \{\{P, A\}, \{0, 1\}, R, P\}$, em que R contém as regras (onde λ é a palavra vazia);

$$P \rightarrow 0P \mid 1A \mid \lambda$$

$$A \rightarrow 1A \mid 1$$

- V. A linguagem L pode ser gerada pela gramática $G = \{\{P, A\}, \{0, 1\}, R, P\}$, em que R contém as regras;

$$P \rightarrow 0P \mid 0A$$

$$A \rightarrow 1A \mid 1$$

Marque a alternativa correta:

- a) Estão corretas apenas as afirmativas I e II.
- b) Estão corretas apenas as afirmativas II e III.
- c) Estão corretas apenas as afirmativas II, III e V.
- d) Estão corretas apenas as afirmativas I, III e V.
- e) Estão corretas apenas as afirmativas II, III e IV.

QUESTÃO 11. Analise as seguintes afirmações:

- I. $\{a^n b^k c^n \mid n, k \text{ são naturais}\}$ é uma linguagem que não pode ser reconhecida por um Autômato Finito;
- II. $\{a^n b^m \mid m, n \text{ são naturais}\}$ não possui autômato de pilha determinístico que possa reconhecê-la;
- III. A linguagem dos números binários pode ser gerada pela gramática $G = \{\{P, A\}, \{0, 1\}, R, P\}$, em que R contém as regras;

$$P \rightarrow 0P \mid 1A$$

$$A \rightarrow 1A \mid 1 \mid 0$$

- IV. Toda linguagem livre de contexto é regular, mas nem toda linguagem regular é livre de contexto.

Marque a alternativa correta:

- a) Estão corretas apenas as afirmativas I e II.
- b) Estão corretas apenas as afirmativas II e III.
- c) Estão corretas apenas as afirmativas II, III e IV.
- d) Estão corretas apenas as afirmativas I e III.
- e) Estão corretas apenas a afirmativa IV.



QUESTÃO 12. Dada a linguagem $L = \{0^n 1^k \mid n = k+2 \text{ e } 0 < k < 5\}$, analise as seguintes afirmações:

- I. L é livre de contexto;
- II. L pode ser demonstrada não ser regular usando o lema do bombeamento;
- III. L é regular;
- IV. L é regular e sua expressão regular equivalente é 0^*111^* .

Marque a alternativa correta:

- a) Estão corretas apenas as afirmativas I e II.
- b) Estão corretas apenas as afirmativas I e III.
- c) Estão corretas apenas as afirmativas III e IV.
- d) Está correta apenas a afirmativa III.
- e) Estão corretas apenas as afirmativas II, III e IV.

PROJETO E ANÁLISE DE ALGORITMOS

QUESTÃO 13. Se a complexidade de tempo de um algoritmo, na análise do pior caso, for $f = O(g(n))$, então, qual das opções abaixo está correta?

- a) O número de passos que o algoritmo efetua, para qualquer entrada, é $\Theta(f(n))$.
- b) A função de custo $f(n)$ tem crescimento assintótico menor ou igual a qualquer limite inferior para o problema.
- c) Se $g = \Omega(f(n))$, então $f = \Theta(g(n))$.
- d) $O(g(n)) \cap \Omega(g(n))$ é o conjunto vazio.
- e) Todas as opções acima são corretas.

QUESTÃO 14. Sobre as técnicas de projeto de algoritmos principais, é correto afirmar que:

- a) O algoritmo recursivo do Quicksort, que realiza ordenação por particionamento, aplica a técnica de divisão&conquista, sendo o melhor algoritmo de ordenação conhecido.
- b) O problema de determinar o n -ésimo número da série de Fibonacci é melhor resolvido usando recursividade.
- c) A técnica de programação dinâmica é uma maneira esperta de aplicar o pensamento recursivo em uma estratégia iterativa com o uso de estruturas de dados adequadas.
- d) A técnica gulosa se baseia na construção da solução passo-a-passo, sempre com a escolha da melhor opção local para se obter o ótimo global na resolução de problemas NP-difíceis.
- e) Todas as alternativas acima estão corretas.

QUESTÃO 15. As fórmulas fechadas exata ou aproximada das expressões matemáticas

$$S(n) = \sum_{i=1}^n i^2 \quad e \quad T(n) = \begin{cases} 1, & p/n = 1 \\ 4T(n/2) + 2, & p/n > 1 \end{cases}$$

, são respectivamente:

- a) $S(n) = \frac{2n^3 + 3n^2 + n}{6}$ e $T(n) = 5n^2 + 2$
- b) $S(n) = \frac{n^3 + n^2 + n}{3}$ e $T(n) = 4(n/2) - 3$
- c) $S(n) = \frac{n(n+1)(2n+1)}{6}$ e $T(n) = \frac{5n^2 - 2}{3}$
- d) $S(n) = \frac{2n^3 + 3n^2 + n}{6}$ e $T(n) = 5n^2 + 2$
- e) Nenhuma das alternativas anteriores.