

Terceiro teste de Algoritmos e Estruturas de Dados

27 de Outubro de 2014

Duração: 20 minutos

Sem consulta

Nome:

N. Mec.:

3.3 1: Qual dos trechos de código seguintes é uma implementação correta da operação **push** numa pilha (*stack*):

- | | |
|---|--|
| a) <pre>void push(double v) { assert(cur_size < max_size); data[cur_size++] = v; }</pre> | c) <pre>double push(void) { assert(cur_size < max_size); return data[cur_size++]; }</pre> |
| b) <pre>void push(double v) { assert(cur_size > 0); data[--cur_size] = v; }</pre> | d) <pre>double push(void) { assert(cur_size > 0); return data[--cur_size]; }</pre> |

3.4 2: Numa lista duplamente ligada, qual dos trechos de código seguintes é uma implementação correta de uma função que conta o número de nós que estão depois do nó fornecido como argumento da função?

- | | |
|--|--|
| a) <pre>int count_after(node *n) { int i; for(i = 0; n->next != NULL; n = n->next) i++; return i; }</pre> | c) <pre>int count_after(node *n) { int i; for(i = 0; n->prev != NULL; n = n->next) i++; return i; }</pre> |
| b) <pre>int count_after(node *n) { int i; for(i = 0; n->next != NULL; n = n->prev) i++; return i; }</pre> | d) <pre>int count_after(node *n) { int i; for(i = 0; n->prev != NULL; n = n->prev) i++; return i; }</pre> |

3.3 3: Qual das seguintes funções pode ser usada para incrementar um índice num *buffer* circular de tamanho *size*?

- | | |
|--|---|
| a) <pre>int inc_index(int i) { return (i + 1 < size) ? i + 1 : 0; }</pre> | c) <pre>int inc_index(int i) { return (i < size) ? i + 1 : 0; }</pre> |
| b) <pre>int inc_index(int i) { return (i + 1 < size) ? 0 : i + 1; }</pre> | d) <pre>int inc_index(int i) { return (i < size) ? i + 1 : size; }</pre> |

- 4.5 **4:** Explique resumidamente o que é uma pilha (*stack*). Não se esqueça de indicar quais são as principais operações suportadas por uma pilha e o que elas fazem.

Resposta:

- 5.5 **5:** Explique resumidamente como pode implementar uma fila (*queue*) a partir de um deque.

Resposta: