

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)

```
void push(double v)
{
    assert(cur_size < max_size);
    data[cur_size++] = v;
}
```

c)

```
double push(void)
{
    assert(cur_size < max_size);
    return data[cur_size++];
}
```

b)

```
void push(double v)
{
    assert(cur_size > 0);
    data[--cur_size] = v;
}
```

d)

```
double push(void)
{
    assert(cur_size > 0);
    return data[--cur_size];
}
```

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)

```
int count_after(node *n)
{
    int i;

    for(i = 0; n->next != NULL; n = n->next)
        i++;
    return i;
}
```

c)

```
int count_after(node *n)
{
    int i;

    for(i = 0; n->prev != NULL; n = n->next)
        i++;
    return i;
}
```

b)

```
int count_after(node *n)
{
    int i;

    for(i = 0; n->next != NULL; n = n->prev)
        i++;
    return i;
}
```

d)

```
int count_after(node *n)
{
    int i;

    for(i = 0; n->prev != NULL; n = n->prev)
        i++;
    return i;
}
```

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

a)

```
int inc_index(int i)
{
    return (i + 1 < size) ? i + 1 : 0;
}
```

c)

```
int inc_index(int i)
{
    return (i < size) ? i + 1 : 0;
}
```

b)

```
int inc_index(int i)
{
    return (i + 1 < size) ? 0 : i + 1;
}
```

d)

```
int inc_index(int i)
{
    return (i < size) ? i + 1 : size;
}
```

- 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: