Home ▶ Exercícios de múltipla escolha (em Inglês) ▶ Capitulo 21

Capitulo 21

This activity contains 19 questions.

1.

Section 21.1 Introduction

21.1 Q1: Which of the following is not a dynamic data structure?

- ○ *Linked list.*
- ○ *Array.*
- ○ *Binary tree.*
- ○ *Stack.*

2.

21.1 Q2: Linked lists allow:

- ○ *d. None of the above.*
- ○ *Insertions and removals only at one end.*
- ○ *Insertions and removals anywhere.*
- ○ *Insertions at the back and removals from the front.*

3.

21.1 Q3: Which data structure represents a waiting line and limits insertions to be made at the back of the data structure and limits removals to be made from the front?

- ○ *Queue.*
- ○ *Stack.*
- ○ *Binary tree.*
- ○ *Linked list.*

4.

Section 21.2 Self-Referential Classes

21.2 Q1: Which of the following is a self-referential object?

○

```
class selfRefer
{
   SelfRefer * x;
};.
```

○

```
class selfRefer1
{
   selfRefer2 x;
};.
class selfRefer2
{
   selfRefer1 x;
};.
```

○
```
class selfRefer
{
   selfRefer x;
};.
```

○
```
a.      class selfRefer
{
   int * x;
};.
```

5.

21.2 Q2: The pointer member in a self-referential class is referred to as a:

- ○ *Connector.*
- ○ *Tie.*
- ○ *Link.*
- ○ *Referrer.*

6.

Section 21.3 Dynamic Memory Allocation and Data Structures

21.3 Q1: The __________ operator takes as an argument the type of object being allocated and returns a __________.

- ○ *delete, reference to an object of that type.*
- ○ *new, pointer to an object of that type.*
- ○ *sizeof, reference to an object of that type.*
- ○ *delete, copy of the object of that type.*

7.

21.3 Q2: Given that the line

```
delete newPtr;
```

just executed, what can you conclude?

- ○ *The pointer newPtr still exists.*
- ○ *The memory referenced by newPtr is released only if it is needed by the system.*

○ *The pointer newPtr only exists if there was an error freeing the memory.*

○ *The pointer newPtr is of type void *.*

8.

Section 21.4 Linked Lists

21.4 Q1: __________ is not an advantage of linked lists when compared to arrays.

○ *No need to allocate extra space, "just in case."*

○ *Efficient insertion and deletion.*

○ *Dynamic memory allocation.*

○ *Direct access to any list element.*

9.

21.4 Q2: For a non-empty linked list, select the code that should appear in a function that adds a node to the end of the list. newPtr is a pointer to the new node to be added and lastPtr is a pointer to the current last node. Each node contains a pointer nextPtr.

○ *lastPtr = newPtr;*
lastPtr->nextPtr = newPtr.

○ *lastPtr->nextPtr = newPtr;*
lastPtr = newPtr.

○ *newPtr->nextPtr = lastPtr;*
lastPtr = newPtr.

○ *lastPtr = newPtr;*
newPtr->nextPtr = lastPtr.

10.

21.4 Q3: What kind of linked list begins with a pointer to the first node, and each node contains a pointer to the next node, and the pointer in the last node points back to the first node?

○ *Doubly-linked list.*

○ *Circular, singly-linked list.*

○ *Circular, doubly-linked list.*

○ *Singly-linked list.*

11.

21.4 Q4: How many pointers are contained as data members in the nodes of a circular, doubly linked list with five nodes?

○ *5.*

○ *8.*

○ *15.*

○ *10.*

12.

Section 21.5 Stacks

21.5 Q1: Which of the following statements about stacks is incorrect?

○ *The last node (at the bottom) of a stack has a null (0) link.*

○ *New nodes can only be added to the top of the stack.*

○ *Stacks are first-in, first-out (FIFO) data structures.*

○ *Stacks can be implemented using linked lists.*

13.

21.5 Q2: A stack is initially empty, then the following commands are performed:

```
push 5
push 7
pop
push 10
push 5
pop
```

Which of the following is the correct stack after those commands (assume the top of the stack is on the left)?

○ *10 5.*

○ *7 5.*

○ *5 10 7 5.*

○ *5 10.*

14.

21.6 Q1: A queue performs the following commands (in pseudo-code):

enqueue 4, 6, 8, 3, 1
dequeue three elements
enqueue 3, 1, 5, 6
dequeue two elements
What number is now at the front of the queue?

○ *5.*

○ *6.*

○ *3.*

○ *4.*

15. *21.6 Q2: A linked list has the functions insertAtFront, removeFromFront, insertAtBack and removeFromBack, which perform operations on nodes exactly as their names describe. Which two functions would most naturally model the enqueue and dequeue operations, respectively, of a queue?*

- ○ *insertAtBack and removeFromBack.*
- ○ *insertAtBack and removeFromFront.*
- ○ *removeFromFront and insertAtFront.*
- ○ *removeFromFront and insertAtBack.*

16. *Section 21.7 Trees*

21.7 Q1: Select the incorrect statement. Binary trees (regardless of the order in which the values are inserted into the tree):

- ○ *Always have multiple links per node.*
- ○ *Always have the same shape for a particular set of data.*
- ○ *Are nonlinear data structures.*
- ○ *Can be sorted efficiently.*

17. *21.7 Q2: If you add the following nodes to a binary search tree in the order they appear (left-to-right):*

6 34 17 19 16 10 23 3

what will be the output of a postorder traversal of the resulting tree?

- ○ *3 10 16 23 19 17 34 6.*
- ○ *10 16 23 19 17 34 3 6.*
- ○ *6 3 34 17 16 10 19 23.*
- ○ *3 6 17 16 10 19 23 34.*

18. *21.7 Q3: Which of the following tasks would a binary search tree not be well suited for?*

- ○ *Sorting.*
- ○ *Searching.*
- ○ *Reversing an unsorted sequence.*
- ○ *Duplicate elimination.*

19. *21.7 Q4: If you have a 1000-element balanced binary search tree, what is the maximum number of comparisons that may be needed to find an element in the tree?*

○ *10.*

○ *8.*

○ *500.*

○ *20.*

Clear Answers / Start Over   Submit Answers for Grading

Answer choices in this exercise appear in a different order each time the page is loaded.