COSC1076 | Advanced Programming Techniques

Revision Questions | Week 09 | Answers

1. A struct is like a class where all of the fields are public be default.
2. Using the basic Linked List ADT that has been used over the past few weeks, implement a ***recursive*** version of the following methods:
   (a) `contains`

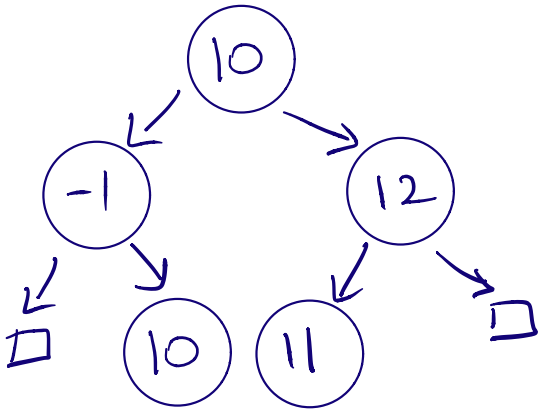
```
bool LinkedList::contains(int value) {
    return contains(value, head);
}
bool LinkedList::contains(int value, Node* node) {
   bool retVal = false;
   if (node == nullptr) {
      retVal = false;
   } else if (node->data == value) {
      retVal = true;
   } else {
      retVal = containts(value, node->next);
   }
   return retVal;
}
```

   (b) `deleteBack`

```
void LinkedList::deleteBack() {
   head = deleteBack(head);
}

Node* LinkedList::deleteBack(Node* node) {
   Node* retVal = nullptr;
   if (node == nullptr) {
      retVal = nullptr;
   } else if (node->next == nullptr) {
      delete node;
      retVal = nullptr;
   } else {
      node->next = deleteBack(node->next);
      retVal = node;
   }
   return retVal;
}
```

3. For a Tree data structure, define the following terms
   (a) Node - Elements of the tree
   (b) Root - First/top node of the tree, that is, the starting point of the tree. Also the node with no parent.
   (c) Leaf - A node with no children
   (d) Child - A subsequent node of any given node
   (e) Parent - The node for which a given node is a child
4. A tree where:
   (a) Each node has 2 children, left and right.
   (b) The value of all left-side child nodes is less than or equal to the node's value
   (c) The value of all right-side child nodes is greater than the node's value.
5. (a) Diagram:



(b) The following sequence of operations will happen:

i. The BST will be deconstructed
ii. The root node of the BST goes out of scope, triggering the deconstructor ot the root node
iii. The left and right child nodes go out of scope, triggering the deconstructors of the left and right child nodes
iv. This process continues until no more nodes go out of scope.