Data Structures Test

1) Items E, F, G and H are inserted into an initially empty structure, in that order, and then one of them is removed. Which is the one removed if (a) the structure is a queue, (b) the structure is a stack?

2) People are queueing to get into a concert. Admission is slow, the queue is long and, from time to time, the people at the back lose heart and drift away. (The people in the middle don't do this as they can't bear the thought that someone who was behind them might just get in.) What structure is this?

3) What is the first thing that an UNSTACK procedure should check for?

4) Suppose that a queue is implemented as an array [1..M]. Fill in the blanks in the following extract from a procedure insertnode:

```
if (rear < M)  /* a */        else  /* b */
```

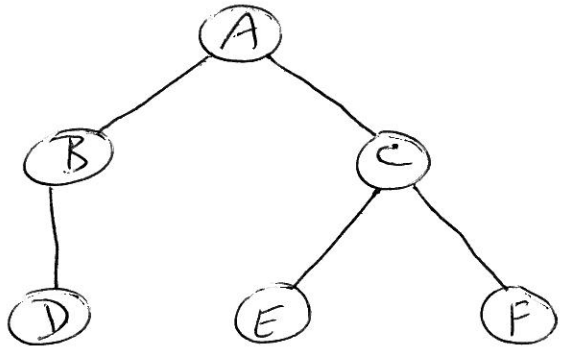
5) Which operation on a general deque is inefficient when the deque is implemented as a singly-linked list, and why?

6) Given a stack implemented as a circular singly-linked list with a pointer PTR, write an algorithm to push a node onto the stack.

7) Draw a diagram illustrating a doubly-linked list with list head after the insertion of the first node.

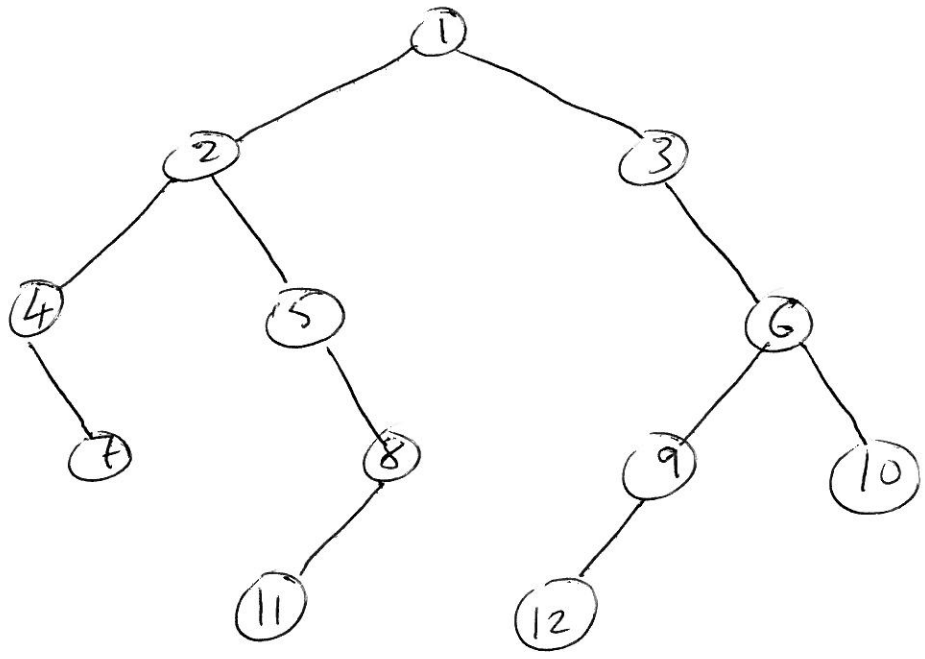
8) Assuming a doubly-linked list with list head, write an algorithm to remove a node pointed at by pointer X.

9) **a)** Is this a binary tree or a general tree?
b) Node A is the _____
c) Node B is at level _____
d) Node C is of degree _____
e) Node E is a _____
f) Node B is the _____ of node D.



10) List the nodes of this binary tree (a) in preorder, (b) in inorder, (c) in postorder.

continued overleaf



11) Redraw the tree in question 10 with the same shape but with the numbers moved around so that it is a binary search tree.

12) Write a recursive procedure to traverse it in postorder.

13) What do you need in order to traverse it non-recursively?

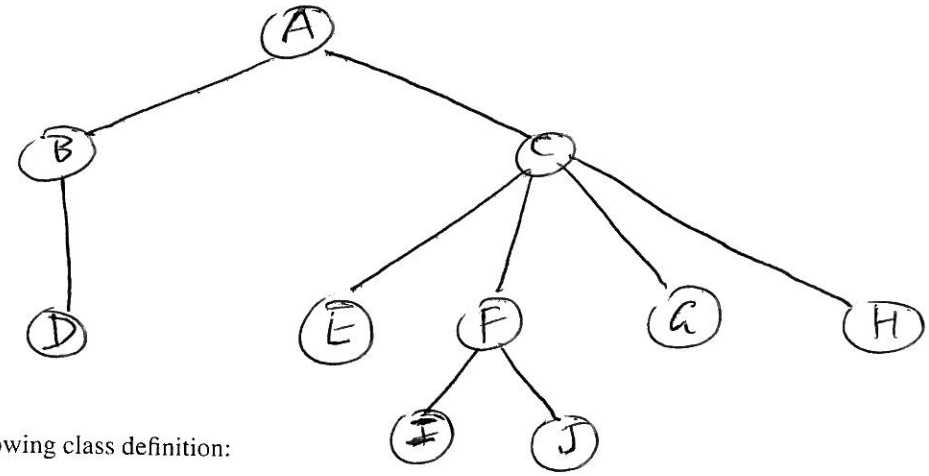
14) How many nil pointers are there in a binary tree?

15) What advantage does a threaded tree have over an unthreaded one?

16) Give a C++ style class definition of a node suitable for a right-threaded binary tree (just the class, not the function definitions).

17) Write an algorithm for inserting a node *X as the right child of node *P in a right-threaded binary tree. (*P may or may not have a right child.)

18) Draw the binary tree that corresponds to this general tree.



19) Assuming the following class definition:

```
class Binode
{ public:
    int  n;
    Binode* l, *r;
};
```

write a recursive procedure in C++ that has two parameters – a pointer to the root node of a binary search tree of integers and a pointer to a newly created node – and which inserts the new node, as a leaf, in its right place.

20) Write a non-recursive version of the procedure described in the previous question.