Feedback — Interview Questions: Stacks and Queues

You submitted this homework on **Thu 17 Sep 2015 9:37 PM EDT**. You will be able to view your score after the deadline passes.

These interview questions are for your own enrichment and are not assessed. If you click the *Submit Answers* button, you will get a hint.

Question 1

Queue with two stacks. Implement a queue with two stacks so that each queue operations takes a constant amortized number of stack operations.

Your Answer	Score	Explanation
Total	0.00 / 0.00	

Question Explanation

Hint: If you push elements onto a stack and then pop them all, they appear in reverse order. If you repeat this process, they're now back in order.

Question 2

Stack with max. Create a data structure that efficiently supports the stack operations (push and pop) and also a return-the-maximum operation. Assume the elements are reals numbers so that you can compare them.

Your Answer	Score	Explanation
Total	0.00 / 0.00	

Question Explanation

Question 3

Java generics. Explain why Java prohibits generic array creation.

Your Answer	Score	Explanation
Total	0.00 / 0.00	

Question Explanation

Hint: to start, you need to understand that Java arrays are *covariant* but Java generics are not: that is, `String[]` is a subtype of `Object[]`, but `Stack<String>` is not a subtype of `Stack<Object>`.

Question 4

Detect cycle in a linked list. A singly-linked data structure is a data structure made up of nodes where each node has a pointer to the next node (or a pointer to null). Suppose that you have a pointer to the first node of a singly-linked list data structure:

- Determine whether a singly-linked data structure contains a cycle. You may use only two pointers into the list (and no other variables). The running time of your algorithm should be linear in the number of nodes in the data structure.

- If a singly-linked data structure contains a cycle, determine the first node that participates in the cycle. you may use only a constant number of pointers into the list (and no other variables). The running time of your algorithm should be linear in the number of nodes in the data structure.

You may *not* modify the structure of the linked list.

Your Answer	Score	Explanation
Total	0.00 / 0.00	

Question Explanation

Hint: maintain a *tortoise* pointer that advances one node per time step and a *hare* pointer that advances two nodes per time step.

Question 5

Clone a linked structure with two pointers per node. Suppose that you are given a reference to the first node of a linked structure where each node has two pointers: one pointer to the next node in the sequence (as in a standard singly-linked list) and one pointer to an arbitrary node.

```
private class Node {
    private String item;
    private Node next;
    private Node random;
}
```

Design a linear-time algorithm to create a copy of the doubly-linked structure. You may modify the original linked structure, but you must end up with two copies of the original.

Your Answer	Score	Explanation
Total	0.00 / 0.00	

Question Explanation

Hint: begin by inserting a new node x' into the linked list immediately after each original node x.