Interview Questions: Stacks and Queues (ungraded) | Coursera

Clip source: Interview Questions: Stacks and Queues (ungraded) | Coursera

Interview Questions: Stacks and Queues (ungraded)

Practice Quiz

Congratulations! You passed!

Grade received 100%

To pass 1% or higher

Interview Questions: Stacks and Queues (ungraded)

Total points 3

1.

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

Note: these interview questions are ungraded and purely for your own enrichment. To get a hint, submit a solution.
1 / 1 point
Keep 2 stacks, let's call them inbox and outbox. Enqueue: Push the new element onto inbox Dequeue: If outbox is empty, refill it by popping each element from inbox and pushing it onto outbox Pop and return the top element from outbox

Correct

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.

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 real numbers so that you can compare them.

1 / 1 point
Don't understand what maximum operation means

Correct

Hint: Use two stacks, one to store all of the items and a second stack to store the maximums.

3.

Java generics. Explain why Java prohibits generic array creation.

1 / 1 point
It's because Java's arrays (unlike generics) contain, at runtime, information about its component type. So you must know the component type when you create the array. Since you don't know what T is at runtime, you can't create the array.

Correct

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>.