Implementation of Queue using two Stacks:

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

Using this method, each element will be in each stack exactly once - meaning each element will be pushed twice and popped twice, giving amortized constant time operations.

Here's an implementation in Java:

public class Queue<E>

{

private Stack<E> inbox = new Stack<E>();

private Stack<E> outbox = new Stack<E>();

public void queue(E item) {

inbox.push(item);

}

public E dequeue() {

if (outbox.isEmpty()) {

while (!inbox.isEmpty()) {

outbox.push(inbox.pop());

}

}

return outbox.pop();

}

}