

In-Class Problems: `queuexfer`

Suppose that a threaded program has N queues of items. The program needs to support an operation called `Transfer(i, j)`. Each call to `Transfer` will transfer a single item from the i th queue to the j th queue, unless there is nothing in the i th queue, in which case the call will not affect the queues.

The program will have multiple concurrent threads, each of which may call `Transfer` zero or more times.

How would you use locks to ensure that `Transfer` operations are atomic? Specifically, how many locks would you use, what would each lock protect, and when would the locks be acquired and released to ensure that transfers are atomic?