10.1-7.

Show how to implement a stack using two queues. Analyze the running time of the stack operations.

Answer.

Implementing a stack is, in other words, maintaining the LIFO property on a sequence of inputs. To manage it with two queues Q_1 and Q_2, we set one of them idle while storing all the stack elements in the other one. When an element is pushed into the stack S, it takes its place at the tail of the nonempty queue. The element pop from the stack S is always the last one in the nonempty queue. To extract this element, we first transfer all its predecessors to the empty queue, then dequeue it. A stack S is said to be empty if both its representaeive queues Q_1 and Q_2 are empty. Conversely, if either Q_1 or Q_2 are full, then S becomes full.

```
STACK-EMPTY(S)
 1  if QUEUE-EMPTY(Q_1) and QUEUE-EMPTY(Q_2)
 2       return TRUE
 3  else return FALSE

STACK-FULL(S)
 1  if QUEUE-FULL(Q_1) or QUEUE-FULL(Q_2)
 2       return TRUE
 3  else return FALSE

PUSH(S, x)
 1  if STACK-FULL(S)
 2       error "overflow"
 3  else
 4       if QUEUE-EMPTY(Q_2)
 5            ENQUEUE(Q_1, x)
 6       else ENQUEUE(Q_2, x)

POP(S)
 1  if STACK-EMPTY(S)
 2       error "underflow"
 3  else
 4       if QUEUE-EMPTY(Q_2)
 5            while Q_1.tail - Q_1.head > 1
 6                 ENQUEUE(Q_2, DEQUEUE(Q_1))
 7            DEQUEUE(Q_1)
 8       else
 9            while Q_2.tail - Q_2.head > 1
10                 ENQUEUE(Q_1, DEQUEUE(Q_2))
11            DEQUEUE(Q_2)
```

PUSH takes $O(1)$ time, while the running time of POP has an order growth $O(n)$.

Email address: informlarry@gmail.com