

hw name

Jacob Purcell, Texas A&M, Student

PROBLEM 1

Assuming a doubly linked list with a head and tail pointer.
To dequeue,

set iterator previous to be the head (previous previous) ($O(1)$)

(18)

delete the head's next node ($O(1)$) (19)

set head's next node to be the iterator ($O(1)$) (20)

have a cursor point to previous node ($O(1)$) (1)

delete information in head pointer ($O(1)$) (2)

set next pointer for cursor to the nullptr ($O(1)$) (3)

make head point to cursor ($O(1)$) (4)

To enqueue,

create a new node with queued information ($O(1)$) (5)

have the next pointer for new node point to tail of queue ($O(1)$)

(6)

have tail point to new node ($O(1)$) (7)

Runtime for dequeue is $4O(1)$ while runtime for enqueue is $3O(1)$, both operate in constant time as $O(1)$ does not depend on the size of the list.

PROBLEM 2

Both methods would be $O(1)$ since it is understood when the queue is made that the next node at the head is the rear and the previous node at the rear is the head.

Enqueue for head iterator;

have new node next point to iterator next ($O(1)$) (8)

have iterator next point to new node ($O(1)$) (9)

Dequeue for head iterator;

have iterator point to it's previous node ($O(1)$) (10)

set iterator next to be the tail (next next) ($O(1)$) (11)

delete the tails previous node ($O(1)$) (12)

set tails previous node to be the iterator ($O(1)$) (13)

Enqueue for tail iterator;

have a new node point to the iterator ($O(1)$) (14)

set new node previous to iterator previous ($O(1)$) (15)

have iterator point to new node ($O(1)$) (16)

Dequeue for tail iterator;

have iterator point to it's next node ($O(1)$) (17)