

Task-1

Onur Yaman
2007961
Homework-05

R-6.8

By 32 enqueue operations, $\text{size} += 32$

By 10 first operations, size does not change

By 15 dequeue operations, $\text{size} -= 15$, but 5 of 15 are raised error, so $\text{size} -= 10$. (When queue is empty, 5 dequeue operations are implemented)

$$32 - 10 = 22$$

$$\text{size} = 22$$

R-6.9

Assuming 32 enqueue operations are implemented before 10 successful dequeue operations, it can be said that size may exceed the capacity 30 if the queue has initially capacity 30

In (circular) array implementation according to lectures, $\text{front} += 10$ since only successful dequeue operations would change the index of front.

Task-2

$S = \text{Array Stack}()$

$Q = \text{Array.Queue}()$

$\text{def pseudo_code}(S, Q, x, r = \text{False})$

$\text{while not } S.\text{is_empty}()$

$\text{if } S.\text{top}() \neq x :$

$Q.\text{enqueue}(S.\text{pop}())$

else:

$r = \text{True}$

break

$\# \text{ Change } _ \text{front} \text{ (front index of } Q)$

$\# \text{ so that } Q._ \text{front} = Q._ \text{size} \text{ in each iteration.}$

$\text{while not } Q.\text{is_empty}()$

$S.\text{push}(Q.\text{dequeue})$

$\text{return } r$

The pseudo-code's time complexity is $O(n)$ since while loops are running $n+1$ times, with all stack and queue operations whose time complexity is $O(1)$ or $O(1)^*$.