Assignment 6

Due: February 24, 2022

Your solutions must be typed (preferably typeset in LaTeX) and submitted as a PDF through Canvas before the beginning of class on the day its due.

Problem 1: Building a Queue using Stacks It is possible to build a *queue* (FIFO) using two stacks. Assume that the stacks have three operations, *push*, *pop*, and *isEmpty*, each with cost 1. Aqueue can be implemented as follows:

- *enqueue:* push item x onto stack 1
- *dequeue:* if stack 2 is empty then pop the entire contents of stack 1 pushing each element in turn onto stack 2. Now pop from stack 2 and return the result.

A conventional worst-case analysis would establish that *dequeue* takes $O(n)$ time, but this is clearlya weak bound for a sequence of operations, because very few dequeues will actually take that long. To simplify your analysis only consider the cost of the push and pop operations.

(a) [10 points] Using the aggregate method show that the amortized cost of each *enqueue* and *dequeue* is constant.

Answer: - In the aggregate method, we consider the cost of all the operations and then take the average of them.

So, if we are considering a sequence of "n" operations, and from the above algorithm, we know that the cost for each element's operation will involve pushing into stack 1, popping from stack 1, pushing into stack 2, and then popping from stack 2.

As we will be following this sequence, the cost will be 4.
So, the total cost of the "n" operations will be T(n) = 4n.

The cost of each operation is 4 i.e. (4n / n) and we can either say that ENQUEUE is taking 4 and DEQUEUE is taking 0, or we can say that the first 3 operations are included for ENQUEUE and the last pop is included for DEQUEUE if this happens.

We can see from the above that the complete operation is having constant amortized cost and will take O(1) time.

References:

https://www.cise.ufl.edu/~sahni/cop5536/exam1/sol3.pdf

(b) [10 points] Using the accounting method show that the amortized cost of each *enqueue* and *dequeue* is constant.

Answer: - In the accounting method, we assign different cost to the operations and then if the cost is more, it is stored as credit for rest of the operations and if it is less, there should be enough credits stored to accomplish it.

So, let us set the amortized cost for both ENQUEUE and DEQUEUE as 4 and 3 respectively.

In case of the ENQUEUE operation, we will be having cost 1 for the Push to stack 1 operation and thus we will store 3 credit for rest of the operations.

Now, for DEQUEUE, we need 3 credits for first stack empty check, last stack empty check and the POP call and then we will use 3 more credits stored with every element cost for stack empty, pushing to stack and popping from stack and therefore we will utilize the 3 stored credits.

From this, we can see that the credit is never negative and cost for each operation (enqueue and dequeue) is a constant value.

(c) [10 points] Using the potential method show that the amortized cost of each *enqueue* and *dequeue* is constant.

Answer: - In the potential method, we assign prepaid work to data structure as a whole instead of assigning credit to individual pieces of the data structure.

Let the potential be Φ for this, and is equal to 2 * number of elements and P(0) is 0 in case of 0 elements.

For ENQUEUE, the actual cost is 1 and change in potential ($\Delta\Phi$) = 2, because if the number of elements increases by 1, change in the potential (P(n) - P(0)) increases by 2, so, by the formula, total amortized cost will be

C = actual cost + $\Delta\Phi$ = 3.

For DEQUEUE, if n is number of elements,
the actual cost is 2n / 2n + 1 (if stack is not empty) and change in potential (0 in case stack is not empty) will be
($\Delta\Phi$) = P(n) - P(0) = 0 - 2n,

so, the amortized cost C = actual cost + $\Delta\Phi$ = 2n - 2n = 0 / 1 (when stack is not empty). From this, we can say that the amortized cost is constant (and greater than 0) for the operations (ENQUEUE and DEQUEUE).

References:

https://www.cise.ufl.edu/~sahni/cop5536/exam1/sol3.pdf