

Auxiliar 6

Análisis amortizado

Profesores: Benjamín Bustos, Gonzalo Navarro

Auxiliares: Sergio Rojas, Pablo Skewes

P1. [P1 - C1 - 2023-2]

Se ha implementado una cola común, con las operaciones **enqueue** y **dequeue**, usando dos pilas, S_1 y S_2 . La idea es que el contenido de la cola sea S_1 leído de arriba hacia abajo, seguido de S_2 leído de abajo hacia arriba. Para encolar ($\text{enqueue}(x)$), se apila x en S_2 ($\text{push}(S_2, x)$). Para descolar ($x = \text{dequeue}()$) se saca x de S_1 ($x = \text{pop}(S_1)$). El problema es que S_1 puede estar vacía. En ese caso, antes del pop, se pasan todos los elementos de S_2 a S_1 : $\text{while not empty}(S_2)$ do $\text{push}(S_1, \text{pop}(S_2))$.

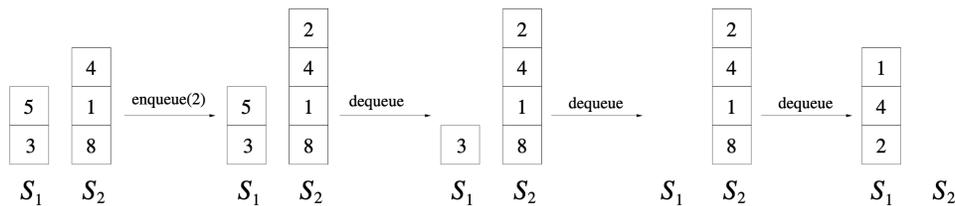


Figura 1: Representación de la cola implementada con dos pilas

Demuestre que esta implementación ofrece tiempo constante amortizado para **enqueue** y **dequeue**. Mida el costo de las operaciones como la cantidad de invocaciones a **push** y **pop**.

P2. [P1 - C2 - 2024-2]

Volviendo al problema del contador binario de k bits visto en clase, donde se demostró que el costo amortizado de la operación **INCREMENTAR** es constante, se desea agregar las operaciones **RESET**, que vuelve a dejar el número binario en 0, y **DECREMENTAR**, que decrementa en 1 el número en caso de no ser cero.

- a) (2pt) La operación **RESET** se implementa manteniendo todo el tiempo la posición max del bit 1 más significativo en el número binario (la posición menos significativa es la 1 y la más significativa es k , con $\text{max} = 0$ si el número binario es cero). Cuando se realiza **INCREMENTAR**, se debe verificar si es necesario incrementar max, y en ese caso hacerlo. En una operación **RESET**, se van cambiando los 1s a 0s desde la posición max a la posición 0, decrementando max en cada caso.

Considere que, al igual que los bit flips, cada incremento o decremento de max cuesta 1. Muestre que el costo amortizado de las operaciones **RESET** e **INCREMENTAR** es $O(1)$ cuando el contador binario parte en 0.

- b) (1pt) Con respecto a la operación **DECREMENTAR**, muestre que una secuencia de operaciones **INCREMENTAR** y **DECREMENTAR** no tiene costo constante amortizado. Para ello, produzca una secuencia de largo m (donde m pueda ser arbitrariamente grande), cuyo costo no sea $O(m)$. Recuerde que consideramos que k no es una constante.