Laboratorio 7

Juan Diego Sique Martínez

Septiembre, 2018

1. Análisis Amortizado utilizando el método de costos acumulativos

1.1. Primer inciso

El problema por defecto nos impone tres métodos a considerar. Los tres métodos a considerar son **PUSH**, **POP** y **MULTIPOP**. Adicional debemos de crear un método que nos permita hacer una copia de seguridad del «stack», nombrado **BACKUP**.

El análisis sin tomar en cuenta ninguna técnica de amortización nos dice que:

POP tendrá un crecimiento asintótico de tipo $O(1)$.

PUSH tendrá un crecimiento asintótico de tipo $O(1)$.

MULTIPOP tendrá un crecimiento asintótico de tipo $O(n)$, donde n reporesenta el número de elementos a retirar del «stack» usando dicha operación.

BACKUP tendrá un crecimiento asintótico de tipo $O(k)$, donde k reporesenta el número de elementos presentes en el «stack».

No obstante, al hacer un análisis acumulativo debemos de modificar los respectivos valores, asumiendo que algunos de ellos representarán un costo menor, y otros mayor, dándonos una diferencia a la que conoceremos como crédito.

Si usamos éste principio podemos decir que añadiremos un valor acumulado de 2 a la operación **PUSH**, considerando que su nuevo costo total es 3. **MULTIPOP** y **POP**, tendrán un costo acumulado negativo, puesto que ya fue considerado en la amortización del **PUSH**, dando un resultado de 0.

$$CostoTotal = CostoIndividual + CostoAgregado \tag{1}$$

$$CostoTotal = 1 + 2 \tag{2}$$

$$CostoTotal = 3 \tag{3}$$

De ésta manera el costo en la operación **BACKUP** será de 0. Ya que con cada operación de **PUSH** estaremos solventando el costo que representa duplicar el arreglo, es decir, por lo menos realizar una operación individual de **PUSH** y **POP**.

Tras establecer ésto último, podemos concluir que hacer k operaciones involucra que haya por lo menos k elementos en el arreglo que contarán con un crédito de $2k$, suficiente para copiar el «stack» y dejar uno vacío para nuevas operaciones. Por lo tanto hacer un número k de operaciones respeta un orden $O(k)$.

1.2. Segundo inciso

El problema consiste en dos operaciones básicas, de las cuales experimentan tiempos muy distintos entre su mejor caso, caso promedio y peor caso. El funcionamiento de una cola utilizando dos pilas funciona así: Se usará la pila primera para ingresar datos a la cola: **ENQUEUE**, y la segunda para extaerlos **DEQUEUE**. La complejidad de las operaciones básicas en su mejor caso es $\Omega(1)$, no obstante, si la pila segunda se encuentra vacía al retirar algún dato la complejidad aumenta a $O(n)$, está en su peor caso, puesto que debe trasladar todos los elementos de la pila primera a la segunda y efectuar la salida del primer dato ingresado.

La cuestión básica es amortizar los costos usando un valor agregado, de manera que su valor asintótico mayor represente un costo constante.

La solución consiste en sumar al costo de la operación **ENQUEUE** que posee un valor unitario [1], un adicional de 3 unidades. La razón de éstas tres unidades es incluir el peor caso en el momento en que la pila segunda se encuentre vacía incluyendo dos operaciones **PUSH** y **POP** que traen consigo el trasladar los elementos de la primera pila a la segunda. Ambas operaciones propias de una pila traen consigo un costo unitario, adicional a estas 2, añadiremos un valor de uno que representa el **POP** propio de la operación **DEQUEUE**.

Por lo tanto los costos son:

ENQUEUE con costo total de 4: 1 + 3, resultando un crecimiento $O(1)$.
DEQUEUE con costo de 0: 1 - 1, resultando un crecimiento $O(1)$.

2. Análisis amortizado usando el método potencial (Φ)

2.1. Primer inciso

El método potencial de amortización de costos utiliza indispensablemente una función $\Phi(x)$, que representa el estado de algo. La función potencia es creciente, por lo tanto a un valor independiente «x» mayor, el resultado $\Phi(x)$ es más alto. Sin ésta propiedad el análisis carece de sentido.

Una vez establecido lo anterior procedo a establecer una función $\Phi(x)$ que satisfaga el resultado deseado, es decir que la complejidad de ejecutar cada operación de manipulación de datos **PUSH** y **POP** obtengan una complejidad $O(1)$.

Mi propuesta es que la función $\Phi(x)$ representa el número de elementos en una pila o «stack», como si estuviese aplicándose la operación en dos estructuras de datos es decir que la ecuación que define el estado es:

$$\Phi(n) = StackReal.tamaño() + Stack.Fantasma.tamaño()$$

Donde el «Stack Fanstama», es la futura copia, asumiendo que desde el momento que se introduce el objeto a la pila, se está haciendo la copia de seguridad.

Para el caso en el que estamos agregando un elemento(**PUSH**), la resolución mátemática es:

$$\hat{C} = C_i + \Delta\Phi(n)$$
$$\hat{C} = C_i + \Phi(n+1) - \Phi(n)$$
$$\hat{C} = C_i + 2 \times n + 2 - 2 \times n$$
$$\hat{C} = C_i + 2$$

Y si el costo individual de un **PUSH** es 1 la ecuación quedará así

$$C_i = 1$$
$$\hat{C} = 1 + 2$$
$$\hat{C} = 3$$

Concluyendo, se sabe que un costo de tres se representa asintóticamente como $O(1)$.

En el caso de **POP**, su ecuación que describe su soto se puede representar así:

$$\hat{C} = C_i + \Delta\Phi(n)$$
$$\hat{C} = C_i + \Phi(n-1) - \Phi(n)$$
$$\hat{C} = C_i + 2 \times -2 - 2 \times n$$
$$\hat{C} = C_i - 2$$
$$C_i = 1$$
$$\hat{C} = 1 - 2$$
$$\hat{C} = -1$$

Y sabemos que un costo negativo en cuestión de análisis asintótico podemos representarlo como 0. Resultando nuestra función **POP**, puede ser representada como $O(1)$.

Por lo tanto al ejecutar «k» operaciones sobre este arreglo:

$$k \times O(n) = O(k)$$

Obteniéndose el resultado deseado.

2.2. Segundo inciso

Al igual que el primer problema, el resultado deseado es poder implementar las operaciones **ENQUEUE** y **DEQUEUE** en un tiempo amortizado de $O(1)$.

La dificultad consiste en encontrar la función Φ que modele correctamente el costo que se amortizará.

La función Φ propuesta es una que representa el número de operaciones que involucra según la estructura. Por lo tanto mi función Φ se define como el número de movimientos **PUSH** y **POP** necesarios para trasladar un elemento de la pila de operación Ξ a la segunda pila. Denótese que la pila de operación es una variable representada por Ξ, que será la pila primera[1] para **ENQUEUE** y la pila segunda[2] para **DEQUEUE**. Ξ usa un coeficiente dos, ya que representa

las dos operaciones que son posibles de hacer en un «stack» o pila: **PUSH** y **POP**

$$\Xi(p) = 2 - \left\lfloor \frac{p}{2} \right\rfloor \times 2$$

$$\Phi(p, n) = \Xi(p) \times n$$

Es decir que para **ENQUEUE** el resultado de la direncia ente funciones Φ será:

$$\Delta\Phi(1, n) = \Xi(1) \times (n + 1) - \Xi(1) \times n$$

$$\Delta\Phi(1, n) = 2 \times (n + 1) - 2 \times n$$

$$\Delta\Phi(1, n) = 2$$

Por lo tanto al calcular su costo total:

$$\hat{C} = C_i + \Delta\Phi(1, n)$$

$$\hat{C} = C_i + 2$$

$$C_i = 1$$

$$\hat{C} = 1 + 2$$

$$\hat{C} = 3$$

De la misma manera se procede a calcular el costo de la acción **DEQUEUE**:

$$\Delta\Phi(2, n) = \Xi(2) \times (n - 1) - \Xi(2) \times n$$

$$\Delta\Phi(2, n) = 0 \times (n - 1) - 0 \times n$$

$$\Delta\Phi(2, n) = 0$$

$$\hat{C} = C_i + \Delta\Phi(2, n)$$

$$\hat{C} = C_i + 0$$

$$C_i = 1$$

$$\hat{C} = 1 + 0$$

$$\hat{C} = 1$$

Resultando un costo total de 1. Para terminar ambas operaciones con costos de 3 y 1 con respecto a **ENQUEUE** y **DEQUEUE** pertenecen al orden de $O(1)$.