Universidad de Chile
Departamento de Ciencias de la Computación
CC4102 - Diseño y Análisis de Algoritmos

Auxiliar 5 - "R-Trees y Análisis Amortizado"

Profesores: Pablo Barceló
Gonzalo Navarro
Auxiliar: ~~Manuel~~ Ariel Cáceres Reyes

16 de abril del 2018

P1. Quack

Un *quack* es una mezcla entre queues y stacks. Puede ser visto como una lista de elementos, escrita de izquierda a derecha, que soporta las siguientes operaciones:

- `QPush(x)` agrega x en el extremo izquierdo.
- `QPop(x)` remueve y retorna el elemento más a la izquierda.
- `QPull(x)` remueve y retorna el elemento m´as a la derecha.

Implemente un *quack* usando tres stacks de modo que cada operación tenga un costo amortizado constante, considerando que estos stacks son capaces de ejecutar las operaciones Push y Pop en tiempo constante. Almacene los elementos sólo una vez en cualquiera de los 3 stacks, y utilice a lo más $\mathcal{O}(1)$ de memoria adicional.

P2. Trits

Considere el siguiente sistema redundante de dígitos ternarios. Un número se representa como una secuencia de *trits*, que, como su nombre lo indica, pueden tomar tres posibles valores: $0, +1 \; o \; -1$. El valor de un número $t_{k-1}, \ldots, t_0$ (donde cada t_i es un *trit*) se define como:

$$\sum_{i=0}^{k-1} t_i 2^i$$

El proceso de incrementar un número de este tipo es análogo al de la operación en números binarios. Se añade 1 al *trit* menos significativo; si el resultado es 2, se cambia el *trit* a 0 y se propaga una reserva hacia el siguiente *trit*. Se repite el proceso hasta que no exista carry. El proceso de decrementar es análogo. Comenzando de 0, se realiza una secuencia de n incrementos y decrementos (sin un orden particular).

Demuestre que, utilizando esta representación, el costo amortizado por operación es $\mathcal{O}(1)$.

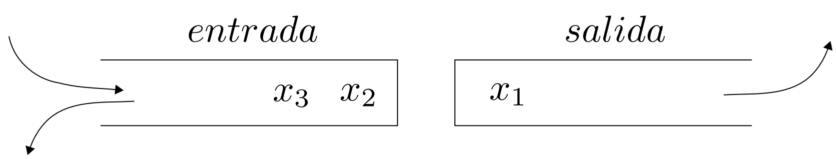
P3. R-Trees

"I save my money, brother. I don't spend what I don't make."
Joe Penny

Soluciones

P1. Quack

Tendremos una pila de *entrada* en donde `qpusheamos` y `qpopeamos` los elementos, y además una pila de *salida* de donde `qpulleamos`. En caso que la pila de *entrada* no tenga elementos para `qpopear`, vaciamos la primera mitad de la pila de salida en una pila *aux*, vaciamos la otra mitad en la pila de *entrada* y vaciamos la pila *aux* en la de salida; llamaremos a esta operación *refilling de entrada* y tiene un costo de $\mathcal{O}(n)$. Por otro lado, en caso que la pila de *salida* no tenga elementos para `qpullear`, ponemos la mitad de los elementos de la pila de *entrada* en una pila *aux*, vaciamos el resto en la pila de *salida* y finalmente vaciamos la pila *aux* en la *entrada*; llamaremos a esta operación *refilling de salida*.



Analizaremos las operaciones por el método de "Contabilidad de Costos". Primero veamos que las tres operaciones tienen costo $\mathcal{O}(1)$ siempre y cuando no se necesite un *refilling*. Por lo tanto, las operaciones pagarán por adelantado estos *refilling*; los `QPUSH` pagarán por adelantado el *refilling de salida* en que pueda participar el elemento pusheado, los `QPULL` pagarán el *refilling de salida* de la mitad correspondiente de elementos que quedaron en la pila de *entrada* en el anterior *refilling de entrada*, finalmente, los `QPOP` pagarán por adelantado el *refilling de entrada* de los elementos correspondientes de la pila de *salida* del anterior *refilling de salida*.

P2. Trits

Consideremos el potencial $\phi =$ cantidad de 1s y -1s. Veamos que $\phi_0 = 0$ y $\phi_i \geq 0$ por ser una cantidad.
Analicemos ahora el costo amortizado por operación.

En el caso del incremento, supongamos que hay l 1s consecutivos en las posiciones menos significativas del número y que luego de ellos hay un 0 o un -1. En este caso tenemos que:

$$\begin{aligned} \hat{c} &= c + \Delta\phi \\ &\leq (l+1) + (-l+1) \\ &= 2 \end{aligned}$$

En el caso del decremento, supongamos que hay l -1s consecutivos en las posiciones menos significativas del número y que luego de ellos hay un 0 o un 1. En este caso también tenemos

"I save my money, brother. I don't spend what I don't make."
Joe Penny

que:

$$\begin{aligned}\hat{c} &= c + \Delta\phi \\ &\leq (l+1) + (-l+1) \\ &= 2\end{aligned}$$

P3. R-Trees

Para más información consultar el apunte del curso.

"I save my money, brother. I don't spend what I don't make."
Joe Penny