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

Auxiliar 4 - "Análisis Amortizado"

Profesor: Gonzalo Navarro
Auxiliar: ~~Manuel~~ Ariel Cáceres Reyes

11 de Septiembre del 2017

P1. Pila con MULTIPOP

Considere una implementación de pila con sus operaciones de `POP` y `PUSH` a costo $\mathcal{O}(1)$. Además considere la operación `MULTIPOP(k)` que remueve los primeros k elementos del tope de la pila y retorna el último de ellos, si la pila tiene $n < k$ elementos entonces se queda vacía. Esta operación tiene costo $\mathcal{O}(\min(n, k))$.
Muestre que el costo amortizado de las operaciones es $\mathcal{O}(1)$ (partiendo de pila vacía).

P2. Cola con 2 Pilas

Con 2 Pilas implemente una Cola costos amortizados de `enqueue` y `dequeue` son de $\mathcal{O}(1)$.

P3. Dinamizando la Búsqueda Binaria

Hacer *búsqueda binaria* sobre un arreglo ordenado toma tiempo *logarítmico* en el tamaño del arreglo, sin embargo la inserción de un nuevo elemento es *lineal* pues se deben "empujar" elementos para mantener el arreglo ordenado.

a) Diseñe una estructura de datos que permita hacer búsqueda en $\mathcal{O}\left((\log n)^2\right)$ e inserción en $\mathcal{O}(\log n)$ amortizado.

b) Explique como se podría implementar la eliminación de elementos a costo $\mathcal{O}(n)$. Muestre además que con esto se pierde el $\mathcal{O}(\log n)$ amortizado de la inserción.

P4. Árboles α-balanceados

Para un valor de $\alpha \in [0.5, 1)$, decimos que un nodo x de un árbol binario es α-balanceado si cumple que:

$$|left(x)| \leq \alpha|x| \quad \wedge \quad |right(x)| \leq \alpha|x|$$

donde $|x|$ es el número de nodos del árbol con raíz en x y $right(x), left(x)$ son los nodos a la derecha e izquierda de x en el árbol. Decimos que un árbol binario de búsqueda es α-balanceado si todos sus nodos son α-balanceado.
Para $\alpha > \frac{1}{2}$ muestre como implementar un árbol binario de búsqueda A α-balanceado cuyos costos de inserción y eliminación sean amortizados *logarítmicos*.

HINTS: Sea lazy, estudie las alturas de estos árboles, estudie los 1/2-balanceados y utilice el potencial $\phi = \frac{1}{2\alpha - 1} \sum_{x \in A; \delta(x) > 1} \delta(x)$, con $\delta(x) = ||right(x)| - |left(x)||$

"Your goal should be to pay off your credit card bills in full at the end of each month and set aside money toward your emergency savings"
Suze Orman

Soluciones

P1. Posible implementación de `MULTIPOP(k)`.

```
1 Funcion MULTIPOP (k)
2 |  for i ← 1...k − 1 do
3 |  |  x ← POP()
4 |  |  if estaVacia() then
5 |  |  |  return x
6 |  |  end
7 |  end
8 |  return POP()
```

Notar que este procedimiento tiene complejidad $\mathcal{O}(\min(n,k))$ y que esto en el peor caso es $\mathcal{O}(n)$. Veremos a continuación que alcanza complejidad amortizada $\mathcal{O}(1)$, en un conjunto de operaciones `POP`, `PUSH` y `MULTIPOP(k)`.

Análisis completo. Supongamos que se hicieron r operaciones de `MULTIPOP` cada una con costo proporcional a $k_1, k_2, \ldots, k_r$ respectivamente. Para poder haber hecho estas operaciones tienen que haber ocurrido al menos (en algún momento) $\sum_{i=1}^{r} k_i$ `PUSH` (para poder hacer esos `MULTIPOP`. Esto significa que en una secuencia de operaciones:

$$\begin{aligned} C_{total} &= C_{PUSHs} + C_{POPs} + C_{MULTIPOPs} \\ &\leq 2C_{PUSHs} + C_{POPs} \\ &\leq 3|numero\ de\ operaciones| \end{aligned}$$

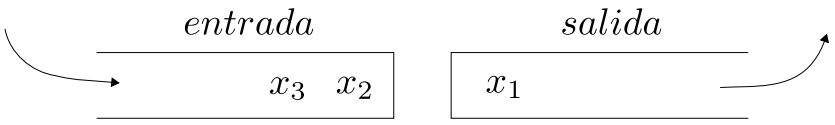
y por lo tanto el costo amortizado de las operaciones es 3.

Contabilidad de Costos. Si le cobramos al `PUSH` 1 por la inserción en el la pila del elemento y 1 por su posible posterior salida de la pila, las operaciones de `POP` y `MULTIPOP(k)` quedan de costo cero, teniendo $\mathcal{O}(1)$ por operción.

Potencial. Si definimos el potencial como $\phi = |Pila|$, tenemos que $\phi_0 = 0$ y además $\phi \geq 0$, entonces $\phi_i \geq \phi_0$, cumpliéndose la condición necesaria para ser un potencial válido.
Usando este potencial, el costo amortizado de `PUSH` queda en $\hat{c} = 1 + 1 = 2$, el de `POP` en $\hat{c} = 1 - 1 = 0$ y el de `MULTIPOP(k)` en $\hat{c} = \min(n,k) - \min(n,k) = 0$.

P2. Tendremos una pila de *entrada* en donde `pusheamos` los elementos cuando ingresan a la cola, y además una pila de *salida* de donde se sacan los elementos de la cola, en caso que esta última no tenga elementos se vacía la cola de *entrada* en la de *salida*.

"Your goal should be to pay off your credit card bills in full at the end of each month and set aside money toward your emergency savings"
Suze Orman



En el caso del `enqueue` es de costo $\mathcal{O}(1)$, sin embargo el `dequeue` es de costo $\mathcal{O}(n)$ pues puede potencialmente ocurrir un vaciado. Veremos a continuación que el costo amortizado de estas operaciones (partiendo de una cola vacía) es $\mathcal{O}(1)$.

Contabilidad de Costos. Al ser encolado, un elemento paga por la operación de `PUSH` en la pila de *entrada* y también paga por su posible posterior traslado de pila (un `PUSH` y un `POP`) por lo que su costo amortizado es de $\hat{c} = 3$. Por lo anterior, el costo de `dequeue` solo es el de hacer `POP` del elemento de la cola de *salida* (pues si hay vaciado, el costo de traslado de pila ya fue pagado por el `enqueue`).

Potencial. Si definimos el potencial como $2|Pila_{entrada}|$, entonces:

- El `enqueue`, $\hat{c} = 1 + 2 = 3$
- El `dequeue`,
  - Sin vaciado, $\hat{c} = 1 - 2 \cdot 0 = 1$
  - Con vaciado, $\hat{c} = 1 + 2|Pila_{entrada}| - 2|Pila_{entrada}| = 1$

P3. a) Tendremos arreglos $A_0, A_1, A_2, \ldots, A_{\lceil \log n \rceil - 1}$ de tamaños $2^0, 2^1, 2^2, \ldots, 2^{\lceil \log n \rceil - 1}$ respectivamente, cada uno de los cuales estará o bien lleno de elementos del conjunto o bien vacío. Además estos arreglos estarán ordenados internamente, pero no existirá alguna relación de orden entre ellos (necesariamente). Si $a_k a_{k-1} \ldots a_1 a_0$ es la representación binaria de n, los arreglos que estarán llenos serán aquellos A_i que tengan a_i en 1 y el resto estará vacío. Notar que esto asegura que tenemos los n elementos en la estructura.

Para buscar un elemento hacemos una búsqueda binaria en cada uno de estos arreglos. Como son $\lceil \log n \rceil$ de ellos, y cada uno es de tamaño $\leq n$ esta búsqueda nos tomará $\mathcal{O}((\log n)^2)$.

Para insertar un elemento buscamos el primer $a_i = 0$ y hacemos la unión de los arreglos $A_{<i}$ con el elemento a insertar, en el arreglo A_i, lo que se puede hacer a costo $\mathcal{O}(2^i)$ que en el peor caso puede ser $\mathcal{O}(n)$.

Análisis Completo. Veamos que para una secuencia de m inserciones, el arreglo A_i será unido $\frac{m}{2^{i+1}}$ veces y por lo tanto el costo agregado de las uniones será

$$\sum_{i=0}^{\lceil \log n \rceil - 1} \frac{m}{2^{i+1}} \cdot 2^i = \lceil \log n \rceil \frac{m}{2}$$

y por lo tanto el costo amortizado será $\mathcal{O}(\log n)$.

b) Para eliminar, buscamos el primer $a_i = 1$. Luego buscamos el elemento que queremos eliminar (supongamos que no está en A_i) y lo intercambiamos con alguno de A_i. Finalmente dividimos A_i en $A_0, A_1, \ldots, A_{i-1}$ a costo $\mathcal{O}(2^i)$ potencialmente $\mathcal{O}(n)$. Si consideramos esta eliminación y la inserción antes descrita obtenemos el siguiente conjunto de operaciones:

1. Insertar elementos hasta que tengamos todo en $|A_i| = 2^i = n$
2. Eliminar un elemento a costo n
3. Insertar un elemento a costo n
4. Repetir 2,3 tanto como sea necesario para que el costo total sea $\sim n^2$

y de este modo las operaciones no pueden tener costo $o(n)$ real y tampoco amortizado.

Lazy Insertaremos/Eliminaremos como en un Árbol de Búsqueda Binaria(ABB), y luego convertimos el nodo más alto que no es α-balanceado en 1/2-balanceado (el subárbol completo). Notar que ser β-balanceado implica ser α-balanceado cuando $\beta \leq \alpha$, por lo que la inserción anterior es correcta.

Estudio de la altura de α-balanceado Notemos que la altura del árbol con n nodos, en el peor caso, será uno más que la altura de un árbol de αn nodos (pues esta es la máxima cantidad de nodos que puede tener alguno de los subárboles hijos). Finalmente resolviendo la ecuación de recurrencia ($h(n) = 1 + h(\alpha n)$) obtenemos que $h(n) \in \mathcal{O}(\log_\alpha n) = \mathcal{O}(\log n)$. Por lo que la búsqueda en estos árboles es logarítmica (y la inserción y eliminación sin reconstrucción).

Estudio de los 1/2-balanceados Para construir un árbol 1/2-balanceado a partir de un ABB cualquiera con raíz x podemos hacer un recorrido *inorden* para obtener los elementos ordenados , luego recursivamente ir eligiendo la mediana del arreglo como raíz (en $\mathcal{O}(|x|)$). El resultado de lo descrito anteriormente es un árbol que cumple que para todo nodo x, $\delta(x) \leq 1$.

Proposición .
Un Árbol es 1/2-balanceado sii todos los nodos del Árbol cumplen con $\delta(x) \leq 1$

$\Rightarrow$
Por contradicción, existe un nodo x tal que $\delta(x) > 1$. Supongamos sin pérdida de generalidad

que $|left(x)| = |right(x)|-2$, por lo tanto se cumple que $|right(x)|-2+|right(x)| = |x|-1$ y por lo tanto $2|right(x)| = |x|+1$, pero por ser 1/2-balanceado se cumple que $|x|+1 \geq 2|right(x)|+1$ lo que es una contradicción.

$\Leftarrow$
En caso que $\delta(x) = 0$ se cumple $|right(x)| = |left(x)| = \frac{|x|-1}{2} \leq \frac{|x|}{2}$. Si en cambio $\delta(x) = 1$, sin pérdida de generalidad $|right(x)| = |left(x)| + 1$ que es igual a $|x| - |right(x)|$ y por lo tanto $|left(x)| < |right(x)| = \frac{|x|}{2}$

Análisis del potencial. Con esto, la reconstrucción propuesta más arriba genera un árbol 1/2-balanceado. Además, el potencial de un árbol 1/2-balanceado es 0.

La inserción y eliminación sin reconstrucción cambian $\delta(x)$ solo de los nodos en el camino de inserción y eliminación a lo más en 1 por cada nodo y por lo tanto el cambio de potencial es $\mathcal{O}(logn)$ obteniendo un costo amortizado de $\mathcal{O}(logn)$.

Por otro lado, supongamos una reconstrucción sobre x que tiene costo $|x|$, el potencial luego de la reconstrucción es 0 (pues queda 1/2-balanceado) y antes de la reconstrucción se cumple $\delta(x) = ||right(x)| - |left(x)|| \geq \alpha|x| - (|x| - \alpha|x|) = (2\alpha - 1)|x|$ y por lo tanto el cambio de potencial es mayor o igual a $|x|$ dejando el costo de la reconstrucción amortizado $\mathcal{O}(1)$.