

Auxiliar 5 - “Análisis Amortizado”

Profesor: Pablo Barceló

Auxiliar: ~~Manuel~~ Ariel Cáceres Reyes

28 de Abril del 2017

P1. Árboles 2-3

Los árboles 2-3 son aquellos árboles de búsqueda en que los nodos internos del árbol pueden contener 1 o 2 elementos y por lo tanto 2 o 3 hijos. En ellos la eliminación puede requerir hacer merge de nodos y la inserción hacer split de los mismos.

- a) Si se realiza un conjunto de inserciones y eliminaciones partiendo de un árbol vacío y n^* corresponde al número máximo de elementos que tuvo el árbol en este proceso, muestre que se puede obtener un costo amortizado de inserciones de $\mathcal{O}(\log n^*)$ y un costo amortizado de eliminaciones de 0. Utilice análisis agregado.
- b) Utilice el método de costos para mostrar que el costo amortizado de inserción es $\mathcal{O}(\log n)$ y el de eliminación es 0, donde n representa el tamaño del árbol en el que estoy realizando la inserción.
- c) Muestre que un conjunto de inserciones, cada inserción realiza $\mathcal{O}(1)$ amortizado splits. Use el método del potencial.
- d) Muestre que en un conjunto de inserciones y eliminaciones no se puede alcanzar un costo amortizado de splits y merges de $\mathcal{O}(1)$.
- e) Muestre que lo anterior se puede lograr en árboles 2-5, en los cuales los nodos pueden tener entre 1 y 4 elementos y por lo tanto, entre 2 y 5 hijos. Utilice el método del potencial.

P2. Doubling and halving

El método de “Doubling and halving” consiste en mantener un arreglo de tamaño variable tal que; cuando este se llena el tamaño del arreglo aumenta al doble y cuando este alcanza la mitad de su capacidad el arreglo se reduce a la mitad de su tamaño.

- a) Muestre que utilizando este método no es posible obtener un costo amortizado $\mathcal{O}(1)$ de las operaciones de insertar y eliminar.
- b) Muestre que si cambio el halving, reduciendo el arreglo la mitad cuando este alcanza un cuarto de su capacidad logra costo amortizado $\mathcal{O}(1)$ en sus operaciones inserción y eliminación. Utilice el método de costos y el del potencial.

P3. Estructura de datos

Diseñe una estructura de datos que mantenga un conjunto de n elementos distintos y soporte las operaciones de inserción en $\mathcal{O}(1)$ amortizado y eliminar la mitad de elementos más pequeña en 0 amortizado. Utilice el método del potencial.

Soluciones

Se usará ca para denotar costo amortizado y cr para denotar costo real.

P1. Árboles 2-3

- a) Podemos demostrar que lo pedido es cierto debemos demostrar que se cumple:

$$\sum_{op} cr(op) \leq \sum_{op} ca(op)$$

Si i es el número de inserciones realizadas y e el número de eliminaciones entonces:

$$\sum_{op} cr(op) \leq i \log n^* + e \log n^*$$

Pero no se pudieron haber hecho más eliminaciones que inserciones por lo que:

$$\begin{aligned} \sum_{op} cr(op) &\leq i 2 \log n^* \\ &= \sum_{op} ca(op) \end{aligned}$$

- b) Por método de costos, cuando realizo una inserción en un árbol de n elementos pago $\log(n+1)$ para una eliminación posterior, por lo que el costo amortizado de la inserción es $(\log n + \log(n+1)) \in \mathcal{O}(\log n)$. Por otro lado, una eliminación en vez de pagar su costo, utiliza el crédito depositado por una inserción anterior, por lo que su costo amortizado es 0.
- c) Definimos el potencial ϕ como el número de nodos con 2 elementos que tiene el árbol. Si definimos k como el número de splits que genera una inserción, veamos que este proceso habrá eliminado k nodos de 2 elementos y quizás creado 1, por lo que:

$$\begin{aligned} ca &= cr + \Delta\phi \\ &= k + -k + 1 \\ &= 1 \end{aligned}$$

- d) Se puede comprobar que si se realiza una inserción en un árbol cuyos nodos de 2 elementos son exactamente aquellos en el camino de la inserción, entonces el resultado de la inserción es un árbol cuyos nodos son todos de 1 elemento y si en este árbol hacemos cualquier eliminación, entonces se generará un camino de nodos de 2 elementos, en el cual podemos hacer luego una inserción como la anterior, repitiendo este proceso tanto como se quiera. Como en el proceso anteriormente descritos se hacen $\log n$ split o merge, tenemos una secuencia de operaciones cuyo costo real es $\log n$ y por lo tanto es imposible que se obtenga un costo amortizado constante de ellas (pues la suma del amortizado debe acotar superiormente al real).

- e) Si ahora consideramos árboles 2 – 5, es decir, árboles que pueden tener entre 1 y 4, la gracia de estos árboles es que al hacer split se generan 2 nodos de 2 elementos (en vez de 2 nodos de 1 elemento) que ayuda a que se obtenga $\mathcal{O}(1)$ amortizado en ambas operaciones. Si definimos el potencial ϕ como la suma de nodos de 1 elemento y nodos de 4 elementos, entonces se cumple que una inserción que genera k splits tiene $\Delta\phi = -k + 1$ (pues se eliminan k nodos de 4 elementos y a lo más se crea uno nuevo) y una eliminación con k merges también tiene $\Delta\phi = -k + 1$ (pues se eliminan k nodos de 1 elemento y a lo más se crea uno de 4), alcanzando con esto $\mathcal{O}(1)$ amortizado de splits y merges.

P2. Doubling and halving

- a) Si consideramos un arreglo a punto de ser llenado e insertamos un elemento, entonces generamos un doubling, si luego de esto eliminamos un elemento generamos un halving, cada uno de los cuales tiene costo $\mathcal{O}(n)$, por lo que si repetimos estas operaciones tanto como queramos es imposible obtener coste $\mathcal{O}(1)$ amortizado de ellas.
- b) Veamos que tanto luego de un doubling como de un halving el arreglo queda a la mitad de su capacidad
- Por lo tanto tiene que hacer al menos $m/2$ (siendo m la capacidad y n el número de elementos) eliminaciones o m inserciones para tener un resize nuevamente. De este modo, haremos que estas operaciones paguen por 2 por el proximo resize, pues si es un doubling necesito m (que los tengo de los $m/2$ pagos de las inserciones), y si hago un halving necesito $m/2$ (que los tengo de los pagos de las $m/4$ eliminaciones anteriores). De este modo alcanzamos un coste amortizado de $\mathcal{O}(1)$ en todas sus operaciones.
 - Si consideramos el potencial:

$$\phi = \begin{cases} 2n - m & n \geq m/2 \\ m - 2n & n < m/2 \end{cases}$$

Se puede comprobar que tanto para las inserciones como las eliminaciones se cumple que $\Delta\phi \leq 2$ por lo que el costo amortizado de estas operaciones es $\mathcal{O}(1)$. Por otro lado, cuando ocurre un resize la situación final siempre deja el potencial en 0 (pues $n = m/2$) y las situaciones iniciales son $\phi = m$ en el caso de doubling y $\phi = m/2$ en el caso de halving que es exactamente el costo de estas operaciones, alcanzando así un costo amortizado 0 y por lo tanto un costo amortizado de todas las operaciones de $\mathcal{O}(1)$.

P3. Estructura de datos

Usamos una lista enlazada para mantener los elementos, lo que nos da inserción en tiempo constante (insertando al final de la lista por ejemplo). Si queremos eliminar la mitad de elementos más pequeña, primero encontramos la mediana en $\mathcal{O}(n)$ (con el algoritmo *mediana de*

medianas visto en cátedras) y hacemos una pasada lineal por la lista eliminando los elementos menores a ella, lo que tiene un costo total de $\mathcal{O}(n)$.

Si la eliminación descrita anteriormente está acotada por cn entonces definimos nuestro potencial como $\phi = 2cn$. De este modo, en una inserción tendremos $ca = cr + \Delta\phi = 1 + 2c$, que da un costo amortizado de $\mathcal{O}(1)$ y al eliminar la mitad inferior $ca = cr + \Delta\phi = cn + (2c(n/2) - 2cn) = 0$, que da un costo amortizado de 0.