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

Auxiliar 4 - "Análisis Amortizado"

Profesores: Pablo Barceló
Gonzalo Navarro
Auxiliar: Dustin Cobas

P1. Una Cola con Dos Pilas

Utilice dos pilas para implementar una cola, donde las operaciones **enqueue** y **dequeue** tengan un costo amortizado $\mathcal{O}(1)$. Considere el costo de dichas operaciones como la cantidad de operaciones **push** y **pop** en las dos pilas usadas.

P2. 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 buscar elementos en un tiempo $\mathcal{O}\left((\log n)^2\right)$ e insertar en un tiempo amortizado $\mathcal{O}(\log n)$.

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

Soluciones

P1. Una Cola con Dos Pilas

Utilizaremos una pila de *entrada* donde añadiremos (`push`) los elementos cuando ingresan a la cola (`enqueue`). La otra pila será utilizada como de *salida*, de la que sacaremos (`pop`) los elementos que se extraen de la cola (`dequeue`). Cuando se solicite extraer un elemento de la cola y la segunda pila esté vacía, pasamos todos los elementos de la pila de entrada a la pila de salida, y extraemos de esta el elemento solicitado.

Contabilidad de Costo Al ingresar un elemento a la cola (`enqueue`), le cobraremos la operación `push` en la pila de entrada, y su posible traslado a la pila de salida (`pop` en la pila de entrada y `push` en la pila de salida), por lo que su costo será de 3 operaciones.

Al extraer un elemento de la cola (`dequeue`) le cobraremos únicamente la operación `pop` en la pila de salida. Note que si esta pila está vacía, el costo de trasladar los elementos desde la pila de entrada ya fue pagado en la operación de `enqueue` en la cola.

Teniendo en cuenta esto, una secuencia de n operaciones en la cola no puede costar más de $3n$ operaciones en las pilas.

Función Potencial Definiremos nuestra función potencial como $\phi = 2 \times |Pila_{entrada}|$.

En el caso de la operación `enqueue`, el costo real $c_i = 1$, por lo que

$$\hat{c}_i = c_i + \Delta\phi_i = 1 + (2 \times |Pila_{entrada}| - 2 \times |Pila_{entrada} - 1|) = 1 + 2 = 3$$

Para la operación `dequeue`, cuando la pila de salida tiene elementos el costo real es $c_i = 1$ y el costo amortizado

$$\hat{c}_i = c_i + \Delta\phi_i = 1 + (2 \times |Pila_{entrada}| - 2 \times |Pila_{entrada}|) = 1$$

Si la pila de salida está vacía, el costo real es $c_i = 2 \times |Pila_{entrada}| + 1$, que es el costo de extraer todos los elementos de la pila de entrada, insertarlos en la pila de salida y extraer el elemento deseado de la pila de salida. El costo amortizado sería entonces

$$\hat{c}_i = c_i + \Delta\phi_i = 2 \times |Pila_{entrada}| + 1 + (2 \times 0 - 2 \times |Pila_{entrada}|) = 1$$

Por lo que el costo amortizado es a lo sumo 3 unidades por operación de la cola.

P2. Dinamizando la Búsqueda Binaria

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.