Departamento de Cs. e Ingeniería de la Computación
Universidad Nacional del Sur

Algoritmos y Complejidad

Trabajo Práctico 6
Análisis Amortizado

primer semestre de 2021

1. ¿Qué podemos afirmar si en un análisis amortizado luego de N operaciones el valor del potencial $\Phi(D_N)$ es menor que el potencial inicial $\Phi(D_0)$?

2. Considere el problema de implementar una cola con dos pilas comunes.

   a) Dar los algoritmos para las operaciones: `ponerEnCola(e)` y `quitarDeCola`. Analizar su tiempo de ejecución.

   b) Dar una función de potencial adecuada y analizar el costo amortizado de cada operación.

3. Considere un arreglo semi-dinámico `A` implementado a través de una lista enlazada `L` donde se almacenan los elementos y un arreglo estático de índices `Ref` que permite el acceso a los elementos de `A`. En general, el elemento almacenado en la posición `i` del arreglo `A`, es el elemento almacenado en la lista `L` en la celda referenciada por `Ref[i]`. En cada celda de `L` se almacena un elemento y el índice del arreglo `Ref` que lo referencia. Se definen las siguientes dos operaciones:

   - `Insertar(elem,i)`: inserta en `A` el elemento `elem` en la posición `i`.
     Si `Ref[i]` no es *nil* entonces cambia el valor del elemento a `elem`.
     Si `Ref[i]` es *nil* entonces agrega una nueva celda al final de `L` (lo cual toma tiempo constante pues siempre se mantiene un enlace a la última celda) y hace las actualizaciones necesarias.
   - `Eliminar(i)`: elimina el i-ésimo elemento de `A`.
     La eliminación consiste solamente en marcar la celda correspondiente con un valor nulo (#). Cuando la cantidad de valores nulos supera la mitad de la cantidad de celdas de L se procede a compactar la lista `L` eliminando las celdas con valores nulos y actualizando el arreglo `Ref` según corresponda.

   a) Escribir los algoritmos para las operaciones `Insertar` y `Eliminar` analizando el orden del tiempo de ejecución de cada uno.

   b) Dar una función de potencial adecuada que permita realizar un análisis amortizado de $O(1)$ para cada operación. Mostrar claramente el análisis amortizado realizado.

4. *Tablas Dinámicas.*
   Sea $\alpha(T_i)$ la *tasa de ocupación* de una tabla dinámica en el momento i, definida como $\frac{\texttt{elementos}(T_i)}{\texttt{capacidad}(T_i)}$. Observar que entonces la expansión ocurre cuando $\alpha(T_i) = 1$.

   a) Mostrar que si se introduce la operación borrado en donde la tabla se contrae si $\alpha(T_i) < 0{,}5$, la función potencial vista en teoría no sirve para demostrar un costo amortizado de $O(1)$ para ambas operaciones.

   b) Considerar una situación idéntica a la anterior, pero en donde la tabla se contrae solamente cuando $\alpha(T_i) < 0{,}25$. Demostrar que la función potencial Φ que se define a continuación permite demostrar un costo amortizado de $O(1)$ para ambas operaciones:

$$\Phi(T) = \begin{cases} 2 \times \texttt{elementos}(T) - \texttt{capacidad}(T), & \text{si } \alpha(T) \geq 0{,}5 \\ \texttt{capacidad}(T)/2 - \texttt{elementos}(T), & \text{si } \alpha(T) < 0{,}5 \end{cases}$$

5. *Skew Heaps (Heaps asimétricos)* [Wei14, sección 11.3]

   a) ¿Qué características tienen los *Skew heaps*?.

   b) Dar un algoritmo para la operación $Mezclar(S_1, S_2)$ que mezcla dos Skew heaps.

   c) ¿A qué se denominan *nodos pesados* y *nodos livianos*? ¿Cómo varía la cantidad de nodos pesados luego de la ejecución de la operación $Mezclar$?

6. *Colas Binomiales*

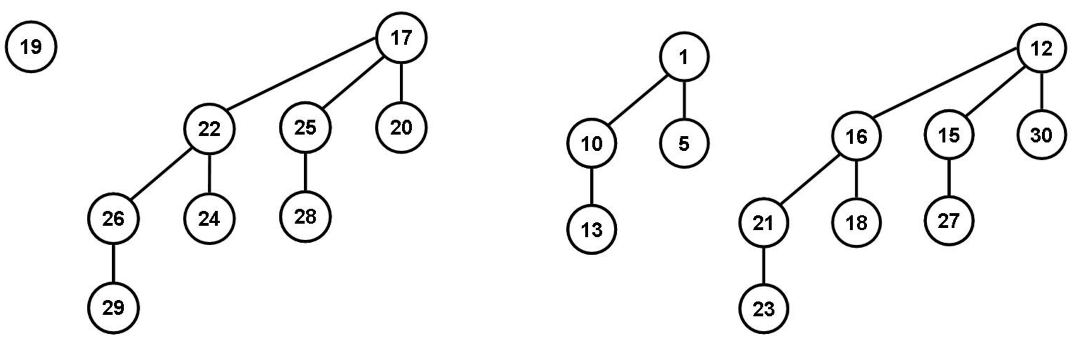
   a) Definir Cola Binomial e indicar cómo implementar en forma eficiente dicha estructura.

   b) Dar un algoritmo para la operación $Minimo(C)$ la cual retorna el mínimo elemento de la cola binomial C.

   c) Dar un algoritmo para la operación $DisminuirClave(C, x, k)$ la cual disminuye el valor de x reemplazándolo por el nuevo valor k.

   d) Dar un algoritmo para la operación $Unir(C_1, C_2)$ la cual une (mezcla) dos colas binomiales. Indicar claramente todos los casos que deben considerarse al momento de proceder con la mezcla.

   e) ¿Cuál es el resultado de realizar la mezcla/unión de las colas binomiales Cb_1 y Cb_2?



(a) Cola Binomial Cb_1 (b) Cola Binomial Cb_2

   f) Indicar cómo se pueden realizar las siguientes operaciones, utilizando las operaciones definidas anteriormente.

   - $Insertar(C, x)$: inserta un nuevo nodo x en la cola C
   - $EliminarMin(C)$: elimina el mínimo elemento de C, y
   - $Eliminar(C, x)$: elimina el nodo x de C,

7. *Fibonacci Heaps*

a) Definir Fibonacci Heap e indicar cómo implementar en forma eficiente dicha estructura. Describir la política del marcado de los nodos y cortes en cascada. ¿Qué sucedería si no se sigue esta política?

b) Describir cómo se realizan las siguientes operaciones.

- $Insertar(F, x)$: inserta un nuevo nodo x en el fibonacci heap F
- $Unir(F_1, F_2)$: crea y retorna un nuevo fibonacci heap que contiene los elementos de los heaps F_1 y F_2 (estos son destruidos como resultado de la operación)
- $Eliminar(F, x)$: elimina el nodo x del heap F.

c) Dar los algoritmos correspondientes para cada una de las siguientes operaciones.

- $EliminarMinimo(F)$: elimina el elemento de F que tiene la clave con menor valor,
- $DisminuirClave(F, x, k)$: disminuye el valor de x reemplazándolo por el nuevo valor k.

d) Analizar cómo varía la cantidad de árboles y la cantidad de nodos marcados luego de la ejecución de cada una de las operaciones que soportan los Fibonacci Heaps.

e) Determinar paso a paso el Fibonacci Heap que se obtiene como consecuencia de la siguiente secuencia de operaciones:

1) Comenzando con un heap vacío, insertar en secuencia los elementos 10, 13, 18
2) Eliminar el mínimo elemento
3) Insertar en secuencia los elementos 14, 16, 20
4) Eliminar el mínimo elemento
5) Insertar en secuencia los elementos 5, 6, 3, 8, 9
6) Eliminar el mínimo elemento
7) Disminuir la clave del nodo 9 al valor 2
8) Unir el heap hasta aquí obtenido con F_1
9) Disminuir la clave del nodo 45 al valor 33
10) Disminuir la clave del nodo 40 al valor 32
11) Eliminar el mínimo elemento

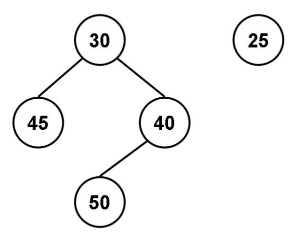


Figura 1: Fibonacci Heap F_1

Referencias

[BB96] Gilles Brassard and Paul Bratley. *Fundamentals of Algorithmics.* Prentice Hall, 1996.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction To Algorithms.* The MIT Press, 3rd edition, 2009.

[Wei14] Mark A. Weiss. *Data Structures and Algorithm Analysis in Java.* Pearson, 3rd. edition, 2014.