

CC30A Algoritmos y Estructuras de Datos – Control 2

Profesor: Patricio Poblete

Junio 10, 2005

1. Suponga que se tiene un heap almacenado en un arreglo $a[1], \dots, a[n]$. Considerando a ese heap como un árbol binario, escriba un algoritmo que liste los elementos de ese árbol en preorden.
2. Suponga que se tiene una tabla de hashing con linear probing almacenada en un arreglo $a[0], \dots, a[m-1]$. Los elementos almacenados son enteros positivos y los casilleros vacíos se reconocen porque contienen ceros. La función de hashing es $h(x) = x \bmod m$.

a) Si la tabla fuera

89	0	42	0	24	54	35	0	0	39
----	---	----	---	----	----	----	---	---	----

indique para cada elemento presente en la tabla cuánto es su costo de búsqueda, e identifique aquél que tiene costo máximo.

- b) Escriba un algoritmo que recorra una tabla de este tipo y encuentre el elemento x que tiene el máximo costo de búsqueda, indicando además cuánto es ese costo.
3. Programe la fase de partición utilizada en Quicksort suponiendo que el pivote se escoge con el método de "mediana de 3".

Más precisamente, programe un método que reciba como parámetros el arreglo a y los subíndices i y j de los dos extremos, y que retorne la posición k en la que queda el pivote al terminar de particionar. Use el método de Lomuto y dibuje claramente el invariante que se mantiene durante la partición. Suponga que el arreglo contiene números enteros. [Recuerde que Lomuto deja a la izquierda a los elementos "chicos", al medio a los "grandes" a la derecha a los "desconocidos"]

Para la muestra con la cual se elige el pivote, considere a los dos elementos extremos ($a[i]$ y $a[j]$) y al elemento que está en el punto medio entre ellos ($a[(i+j)/2]$).

4. Escriba un método que determine si un árbol binario dado es o no AVL. El método debe recibir un puntero a la raíz del árbol y retornar la altura del árbol (si es AVL) o un -1 (si no es AVL). Suponga que un árbol vacío (puntero nulo) tiene altura cero.

Tiempo: 2 horas

Entregar en hojas separadas

Con apuntes de clases