Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Ayudantía 2 Repaso I1

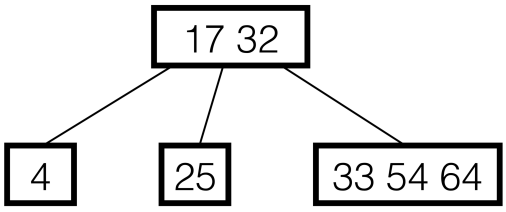
IIC2133 - Estructuras de datos y algoritmos

Segundo semestre, 2017

1. Árboles

1. Dibuja un árbol AVL, preferiblemente pequeño, tal que las tres próximas inserciones tengan que corregirse con rotaciones, tanto simples como dobles.

2. Considere el siguiente árbol 2-4:



   2.1. Realice las siguientes inserciones:
   - Insertar 11
   - Insertar 40
   - Insertar 50
   - Insertar 45

   2.2. Transforme el árbol resultante en un árbol rojo-negro.

   2.3. En el árbol rojo-negro, realice las siguientes inserciones:
   - Insertar 20
   - Insertar 70
   - Insertar 60

   2.4. Haga la transformación de vuelta a un árbol 2-4

2. Ordenación

1. ¿Cuál es la complejidad de `Counting Sort` y `Radix Sort`, `String Sort`? ¿Son algoritmos estables?

2. Supongamos que en `Partition` elegimos el dato en la posición del medio del arreglo como pivote, en vez de elegir el dato en el extremo derecho. ¿Es ahora menos probable que `Quick Sort` requiera tiempo cuadrático? Justifica.

3. ¿Es correcto que si la rutina `Partition` siempre realiza el mínimo número de intercambios posibles, se garantiza que `QuickSort` ejecutará en el menor tiempo posible (es decir, esta en el mejor caso)? Argumente.

3. Ejercicios Propuestos

1. Dado un arreglo desordenado con n números, se desea obtener los números que se encuentran entre las posiciones i e $i + k$ (no necesariamente ordenados). Proponga un algoritmo que resuelva este problema y calcule su complejidad con respecto a n, i y k. Demuestre que no es posible hacerlo en una complejidad menor.

2. Considera un árbol AVL inicialmente vacío, en el que queremos almacenar las claves 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. ¿Es posible insertar estas claves en el árbol en algún orden tal que nunca sea necesario ejecutar una rotación? Da un ejemplo si crees que se puede o una demostración en el caso contrario.

3. Escriba el pseudocódigo de una versión inestable (pero correcta) de `Merge Sort`. Ahora diga cómo transformarla en estable.

4. Indique los pasos a seguir para transformar un árbol rojo-negro en un árbol 2-4.

5. Suponga que insertará los objetos $o_1, o_2, ..., o_n$, en orden, a un `Min-Heap`, usando la operación de inserción estándar. ¿Cómo deberían ser las claves de estos objetos para obtener el mejor caso? ¿Y para obtener el peor caso? Diga, con precisión, cuántos intercambios se producen en cada caso si se usan las rutinas vistas en clases.

Link útil

`https://www.cs.usfca.edu/~galles/visualization/Algorithms.html`