



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2133 Estructuras de datos y algoritmos
1^{er} semestre 2017

Ayudantía de Árboles

1. Árboles binarios de búsqueda

1. Escriba en pseudocódigo como encontrar el predecesor de una clave k dentro del árbol, comente la complejidad del algoritmo en términos del número de claves n .
2. Supongamos que cada vez que nuestro programa tiene acceso a una clave k de un árbol, necesita saber cuántas claves en el árbol son menores que k . Esto se conoce como el rango de k .
 - a) ¿Qué información adicional sería necesario almacenar en el árbol?
 - b) ¿Cómo se determinaría el rango de k y cuál sería la complejidad de esta operación?
 - c) ¿Cuánto costaría mantener actualizada la información adicional del árbol cuando se produce una inserción de una clave?

2. Árboles AVL (Balanceados)

1. Ahora que sabemos que el árbol está balanceado, ¿Cuál es la complejidad de encontrar el predecesor?
2. ¿Cuál es el tamaño de la rama más corta que puede tener un árbol AVL? Justifique.
3. Imagina que se conoce a priori las claves que serán insertadas en el árbol, ¿Existe algún orden de inserción tal que nunca sea necesario hacer una rotación?, asuma que tiene las llaves 1,2,3,4,5,6,7.

3. Árboles 2-4

1. Estos árboles tienen la propiedad de ser perfectamente balanceados. ¿Qué significa esto?. Muestre esta propiedad.
2. Hay casos en que la operación de inserción no necesita hacer splits a pesar de que la inserción cause overflow sobre un nodo. ¿Cuál son estos casos y como evitaría los splits?. HINT: Vea los vecinos/hermanos del nodo.