



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2133 - ESTRUCTURAS DE DATOS Y ALGORITMOS

Ayudantía 2

24 de abril de 2020

Árboles de Búsqueda Binarios (ABB)

Pregunta 1 [I1 2014-1 P3]

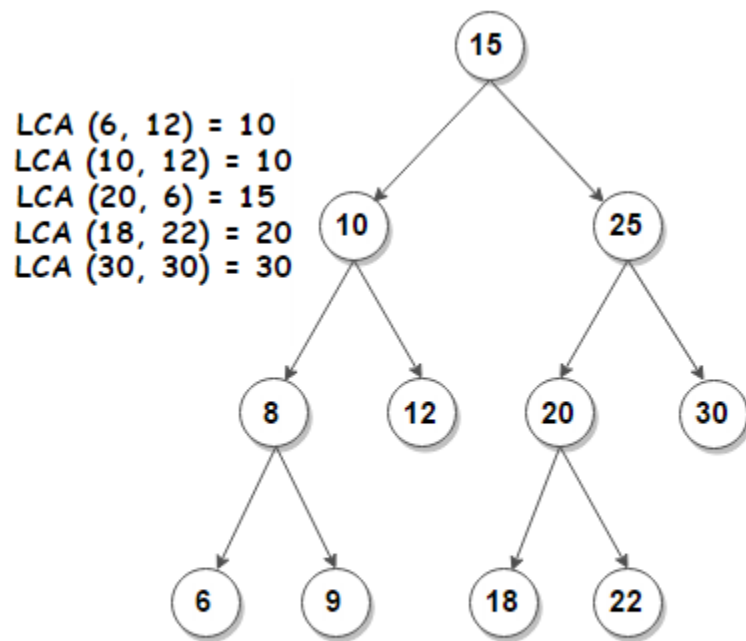
- Supongamos que tenemos una estimación por adelantado de cuán a menudo nuestro programa va a tener acceso a las claves de los datos (es decir, conocemos la frecuencia de acceso a las claves). Si queremos emplear un *ABB*, inicialmente vacío, para almacenar los datos (según sus claves), ¿en qué orden deberíamos insertar las claves en el árbol? Justifica.
- Supongamos que cada vez que nuestro programa tiene acceso a una clave k , necesita saber cuántas claves en el árbol son menores que k . Esto se conoce como el rango de k . ¿Qué información adicional sería necesario almacenar en el árbol? ¿Cómo se determinaría el rango de k y cuál sería la complejidad de esta operación? ¿Cuánto costaría mantener actualizada la información adicional del árbol cuando se produce una inserción o una eliminación de una clave?

Pregunta 2 [Propuesta]

El *lowest common ancestor* (LCA) [más bajo común antecesor] entre dos nodos x e y en un ABB, es el nodo más bajo (profundo) que tiene tanto a x como a y como descendientes, donde cada nodo puede ser descendiente de sí mismo (si x es alcanzable desde y , entonces el LCA es y).

En otras palabras, el LCA entre x e y es el antecesor compartido (incluidos ambos como opciones) que está más lejos de la raíz.

Describe la lógica de cómo sería un algoritmo eficiente en términos de tiempo y de memoria, para encontrar el LCA entre dos nodos x e y .



Lowest Common Ancestor in BST

AVL

Pregunta 3

- ¿ Es todo árbol AVL un árbol rojo-negro? Justifique o dé un contraejemplo. (I2 2015-2)
- Describe un algoritmo de certificación para árboles AVL, suponiendo que sabemos que el árbol es un ABB. Es decir, tu algoritmo debe verificar que la propiedad de balance del árbol se cumple y que la información de balance mantenida en cada nodo esté correcta. Tu algoritmo debe correr en tiempo $O(\text{número de nodos del árbol})$ (I1 2014-1)
- Propuesto: demuestra la correctitud y finitud del algoritmo anterior