

Auxiliar 8 - Más árboles

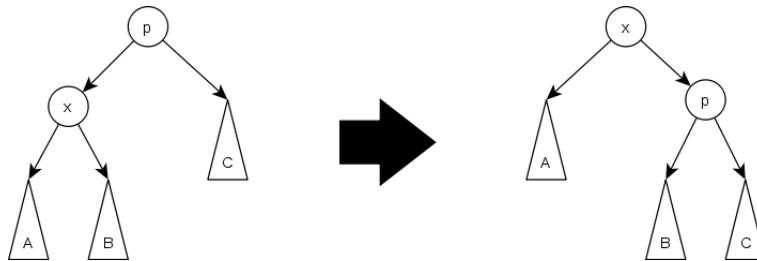
Profesores: Nelson Baloian
Jeremy Barbay
Patricio Poblete

Auxiliares: Gabriel Flores, Sven Reisenegger
Juglar Díaz, Gabriel Norambuena
Cristóbal Muñoz

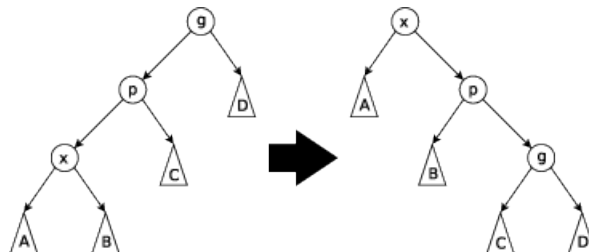
P1. Splay Trees

Un *splay tree* (o árbol biselado) es otro tipo de árbol de búsqueda binaria autobalanceante, cuya propiedad más interesante es que los últimos nodos buscados e insertados están más cerca de la raíz que los nodos más antiguos. En particular, el último nodo insertado/accedido x va a ser la nueva raíz del árbol. Esto se logra mediante una operación llamada *splaying* sobre x . Hay tres casos para ejecutar esta operación:

- **Caso 1: zig** *El padre de x es la raíz*
Se hace una rotación simple sobre el padre (al igual que en árboles AVL y rojinegros), dejando a x como la nueva raíz.

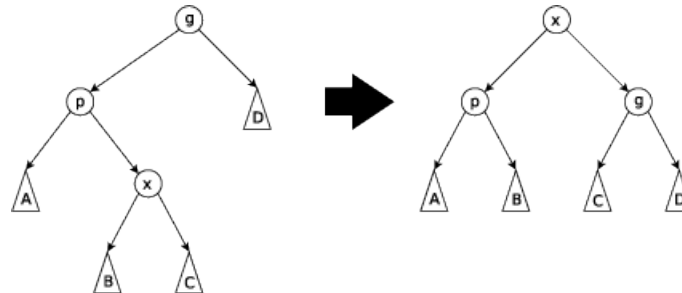


- **Caso 2: zig-zig** *Tanto x como su padre son hijos del mismo lado (izquierdos o derechos) de su abuelo*
Se hacen dos rotaciones simples: primero entre abuelo y padre, y luego entre x y su padre.



- **Caso 3: zig-zag** *El nodo x es hijo del distinto lado del que su padre lo es de su abuelo (están en zigzag)*

Se hace una rotación simple entre x y su padre. Luego, rotar como en el caso 1 (en la imagen, el árbol $(p \ A \ B)$ cumple el rol del árbol A resultante en el diagrama del caso 1).



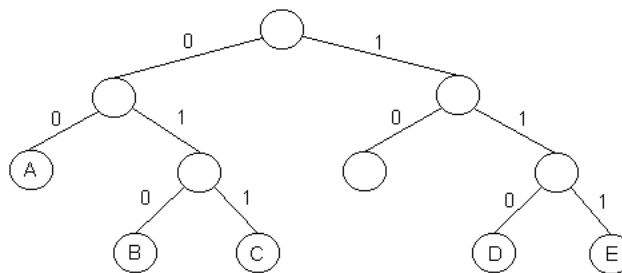
- Inserte en un *Splay Tree* vacío la siguiente secuencia: 20, 7, 12, 16, 2. Luego, acceda al nodo con valor 7, subiéndolo a la raíz con las operaciones de *splay* como corresponda. Comente acerca de la complejidad de la búsqueda en promedio.
- Inserte en un *Splay Tree* vacío la siguiente secuencia: 1, 2, 3, 4, 5, 6, 7, 8. Luego, acceda al nodo con valor 8, realizando el *splaying* correspondiente. Comente acerca de la complejidad de búsqueda en este caso en particular.

P2. Tries

Un *trie* (o árbol digital) es un tipo de árbol de búsqueda (no necesariamente binaria), que tiene la particularidad de no guardar valores *en* sus nodos, sino que la información está dada por la posición que tiene un nodo dentro del árbol. La búsqueda se realiza desde la raíz, eligiendo la i -ésima arista a recorrer según i -ésimo carácter del valor que se desea buscar. Note que la búsqueda tarda $O(\text{largo de la palabra buscada})$.

Codificación:

A = 00100
B = 01000
C = 01111
D = 11000
E = 11101



- Inserte en un *trie* vacío las palabras PPRPOQ, QOQOQQP, PPRPPQ, PRRPQQO, QOQQPPR, PPRPPQ, PPRPQO. ¿Cuál es el prefijo más largo que se repite?
- Inserte ahora la clave PPRPQOO. Comente sobre la implementación.