



Act 3.3 - Investigación de Splay Tree

Programación de estructuras de datos y algoritmos fundamentales (Gpo 4)

Alumnos:

Thomas Freund Paternostro //A00831997

Fecha de entrega:

13/10/2021

¿Qué es Splay Tree?

Es otra variante del árbol binario y fue inventado por Daniel Sleator y Robert Tarjan en 1985. Este es un BST que se auto-ajusta o arregla de tal manera que cada operación en un elemento permita que el elemento se encuentre en la raíz del árbol. Es un algoritmo que tiene un espacio $O(n)$ y un tiempo $O(\log n)$ en el peor de los casos.

¿Cómo se ocupa Splay Tree?

En este tipo de BST, cada operación realizada se realiza en la raíz del árbol y esta operación para este árbol en particular se conoce como “spalying”. Splaying se puede definir como el proceso en traer a la raíz el elemento que se desea trabajar al realizar las operaciones necesarias para moverlo ahí.

Al realizar un spaly en un elemento, se trae elementos frecuentemente usados cerca de la raíz de tal manera que cualquier operación realizada en esos elementos sea rápida. Por lo tanto los elementos menos frecuentes están más lejos de la raíz y los más frecuentes son los más cercanos de igual manera.

En el caso de la operación de inserción, o realmente cualquier otra usado en un BST, primero se inserta un método utilizando el proceso de inserción de BST regular, pero a este elemento se le aplica un spray de tal forma que se encuentra en la raíz.

¿Para qué se ocupa Splay Tree?

Se ocupa para poder acceder de manera rápida e inmediata a los elementos más frecuentes al traerlos a la raíz del árbol. Dado que los elementos frecuentes están cerca de la raíz estos podrán ser accedidos en un tiempo $O(1)$.

Un uso práctico de un splay tree para entender para que se ocupa sería un red de router. Un router recibe paquetes de información constantemente de diferentes conexiones y debe decidir en cuál cable de salida debe mandar ese paquete basado en el IP del mismo. El router para poder hacer esto hace de una mapa de IPs para poder realizar esta conexiones y si un IP fue utilizado una vez este es más propenso a ser llamada continuamente y en lo ideal es poder hacerlo en el menor tiempo posible y un splay tree es una estructura de datos que facilita procesos como este.

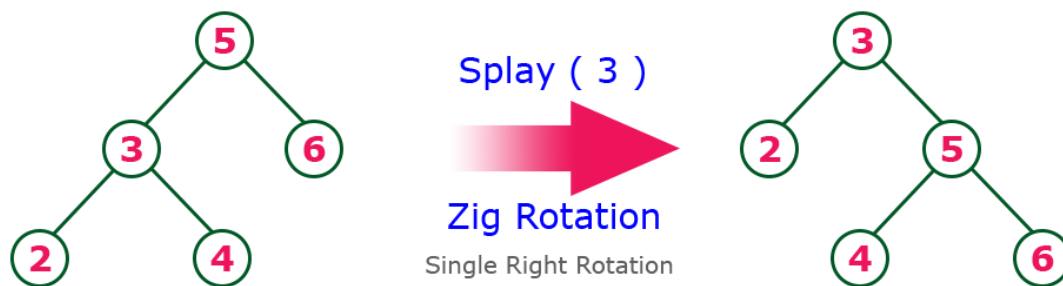
¿Cómo se implementa Splay Tree?

Un splay tree se enfoca en realizar diferentes movimientos para realizar un splay.

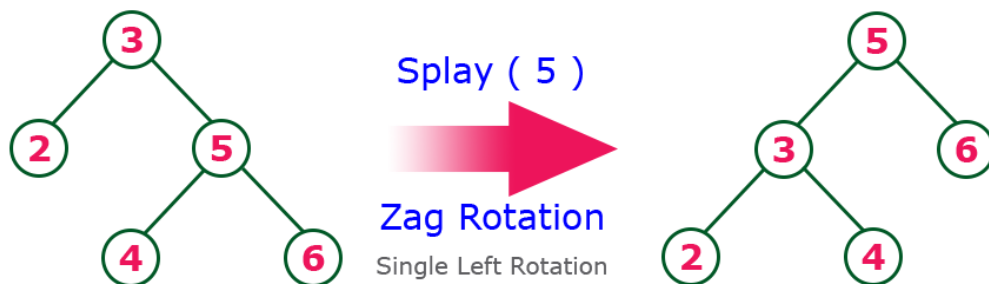
Rotaciones:

“La **rotación en zig** en el árbol de distribución es similar a la rotación única a la derecha en las rotaciones del árbol AVL. En la rotación en zig, cada nodo se mueve a una posición a la

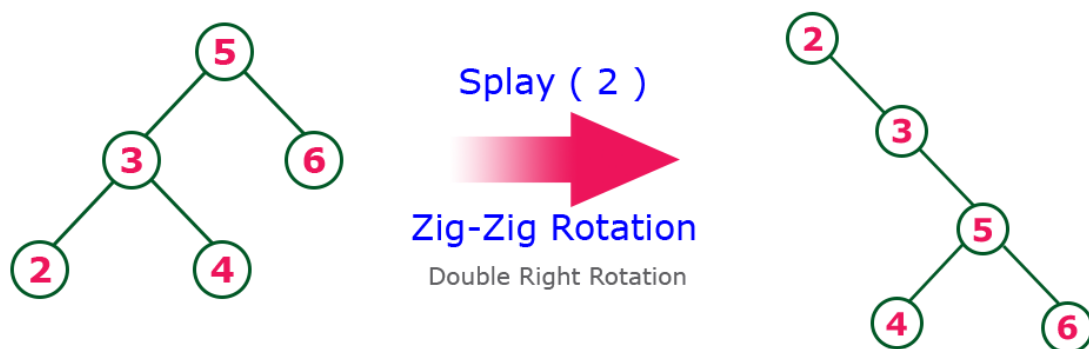
derecha desde su posición actual (Albers,2002).”



“La rotación **Zag** en el árbol de distribución es similar a la rotación única a la izquierda en las rotaciones del árbol AVL. En la rotación zag, cada nodo se mueve a una posición a la izquierda desde su posición actual (Albers,2002).”



“La rotación en **zig-zig** en el árbol de distribución es una rotación en doble zig. En la rotación en zig-zig, cada nodo se mueve dos posiciones a la derecha desde su posición actual (Albers,2002).”



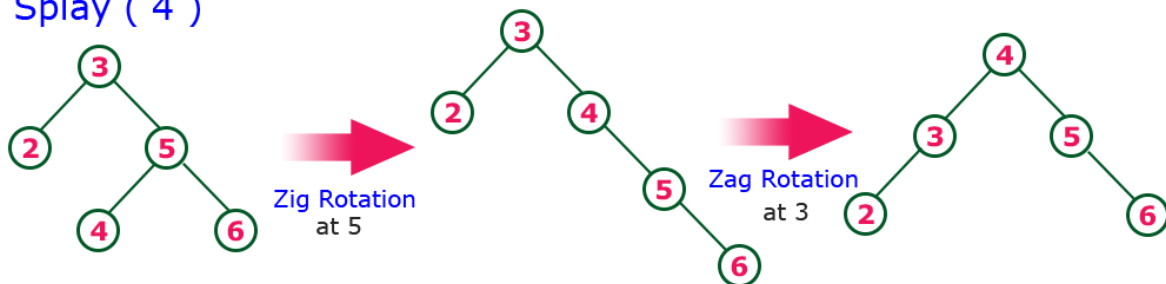
“La rotación **Zag-Zag** en el árbol de splay es una rotación de doble zag. En la rotación zag-zag, cada nodo se mueve dos posiciones hacia la izquierda desde su posición actual

(Albers,2002).”

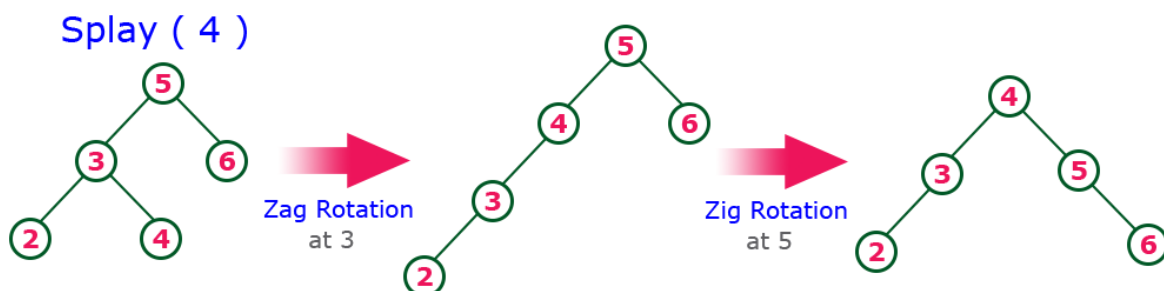


“La rotación en zig-zag en el árbol de distribución es una secuencia de rotación en zig seguida de rotación en zag. En la rotación en zig-zag, cada nodo se mueve a una posición a la derecha seguida de una posición a la izquierda desde su posición actual (Albers,2002).”

Splay (4)



“La rotación Zag-Zig en el árbol de splay es una secuencia de rotación en zag seguida de una rotación en zig. En la rotación zag-zig, cada nodo se mueve a una posición a la izquierda seguida de una posición a la derecha desde su posición actual (Albers,2002).”



Nota: Todo splay tree es un BST pero no necesita ser un árbol balanceado o un balanced tree.

La operación de inserción :

“La operación de inserción en el árbol Splay se realiza mediante los siguientes pasos :

Paso 1: Comprobar si el árbol está vacío.

Paso 2: Si el árbol está vacío, insertar el nuevo nodo como nodo raíz y salga de la operación.

Paso 3: Si el árbol no está vacío, insertar el nodo nuevo como nodo hoja utilizando la lógica de inserción del árbol de búsqueda binaria (BST).

Paso 4: Después de la inserción, extienda el newNode (Albers,2002).”

La operación de eliminación:

“La operación de eliminación en el árbol de splay es similar a la operación de eliminación en el Árbol de búsqueda binaria. Pero antes de eliminar el elemento, primero debemos expandir ese elemento y luego eliminarlo de la posición raíz. Finalmente, únase al árbol restante utilizando la lógica del árbol de búsqueda binaria (Albers,2002).”

Referencias

Albers, S. (2002). Randomized splay trees: Theoretical and experimental results.

Information Processing Letters, 81(4), 213–221.

[https://doi.org/10.1016/s0020-0190\(01\)00230-7](https://doi.org/10.1016/s0020-0190(01)00230-7)

Rajinikanth, B. (s. f.). *Data Structures Tutorials - Splay Tree with an example*.

Btechsmartclass. Recuperado 14 de octubre de 2021, de

http://www.btechsmartclass.com/data_structures/splay-trees.html