

Auxiliar 7 - Heap y Árboles

Profesores: Jérémy Barbay
Patricio Poblete, Nelson Baloian
Auxiliares: Felipe Lizama, Franco Sanguinetti,
Matías Ramírez, Sven Reisenegger.

P1. Checkear AVL

Un árbol AVL es un tipo especial de ABB auto-balanceable, en que todos sus nodos cumplen con la condición de que la diferencia de alturas entre su hijo izquierdo y derecho nunca es mayor que uno.

De manera formal, un árbol T , con hijos T_i y T_d es AVL si cumple con las siguientes condiciones:

- $|altura(T_i) - altura(T_d)| \leq 1$
- T_i es AVL
- T_d es AVL

Escriba una función `es_avl(raiz)` en **Python**, que dada la raíz de un árbol diga si éste cumple con las condiciones para ser AVL o no.

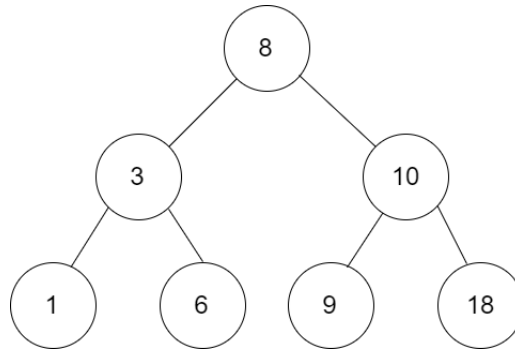
P2. Min Heap

Un *heap* es un árbol binario que permite almacenar valores sin la necesidad de utilizar punteros (utilizando sólo un arreglo). En el caso del *min heap*, su principal característica es que el padre cuenta siempre con menor prioridad que sus hijos, esto es, el mínimo se encuentra en la raíz. Cuenta con todos sus niveles llenos a excepción de probablemente el último. Además, en el último nivel, los nodos se encuentran lo más a la izquierda posible. Un *heap* se utiliza para implementar una cola de prioridad.

Considere la siguiente secuencia de números: 95,45,73,40,33,60,38,15,10,20,52,22

- (a) Tome los nodos de a uno e insértelos en el *min heap*. Dibuje el estado del arreglo en cada inserción.
- (b) Extraiga tres veces el mínimo y reordene el *heap* de tal forma que siga manteniendo forma y orden de *heap*.
- (c) Dibuje el árbol representado por el *heap*
- (d) Implemente una cola de prioridad utilizando un *heap* en **Python**, que sea capaz de agregar un elemento y de retornar el mínimo.

P3. K-ésimo elemento de un ABB



Dado un puntero a la raíz de un árbol de búsqueda binaria, se quiere retornar su k -ésimo menor elemento. Por ejemplo, en el caso de la imagen, si k vale 3, se debe retornar 6 y si k es igual a 5, se debe retornar 9. Para realizar esto, cree una clase `NodoArbol` de manera que cada nodo sea capaz de guardar la cantidad de nodos con los que cuenta su subárbol izquierdo. Además programe la función `insertar(raiz, valor)` que inserte un elemento en el ABB y mantenga dicha cantidad actualizada. Utilice la cantidad de nodos a la izquierda para diseñar un algoritmo que permita encontrar el k -ésimo elemento en tiempo $O(h)$ en donde h es la altura del árbol.