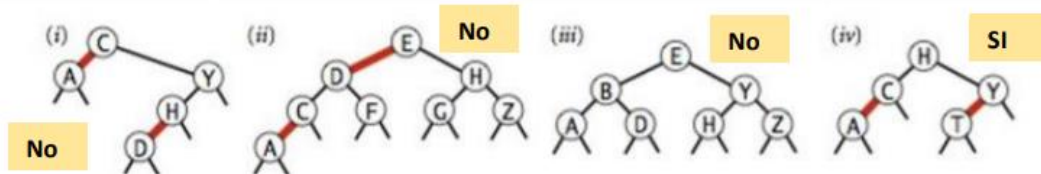


Control de lectura

Integrantes:

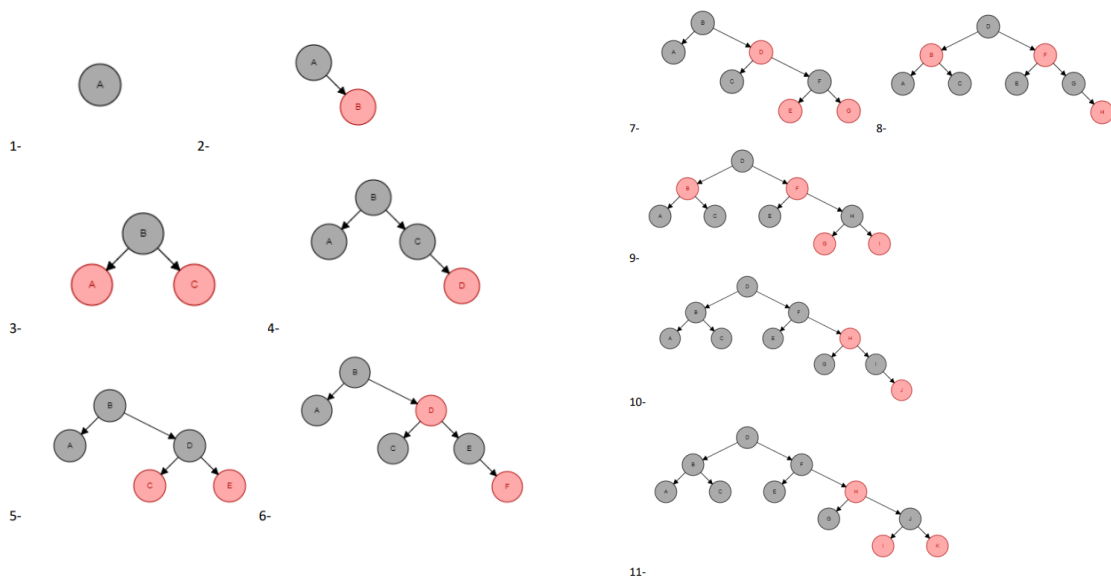
- Diego Cárdenas
- Felipe Calvache
- Zayra Gutiérrez

1. ¿Cuál de los siguientes árboles son red-black BSTs?

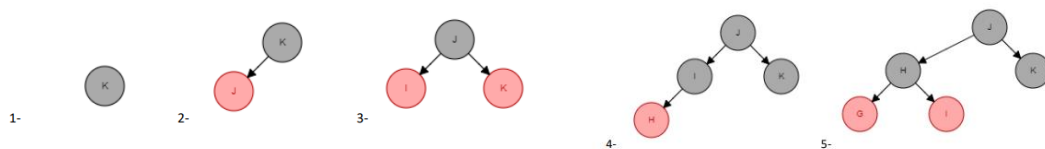


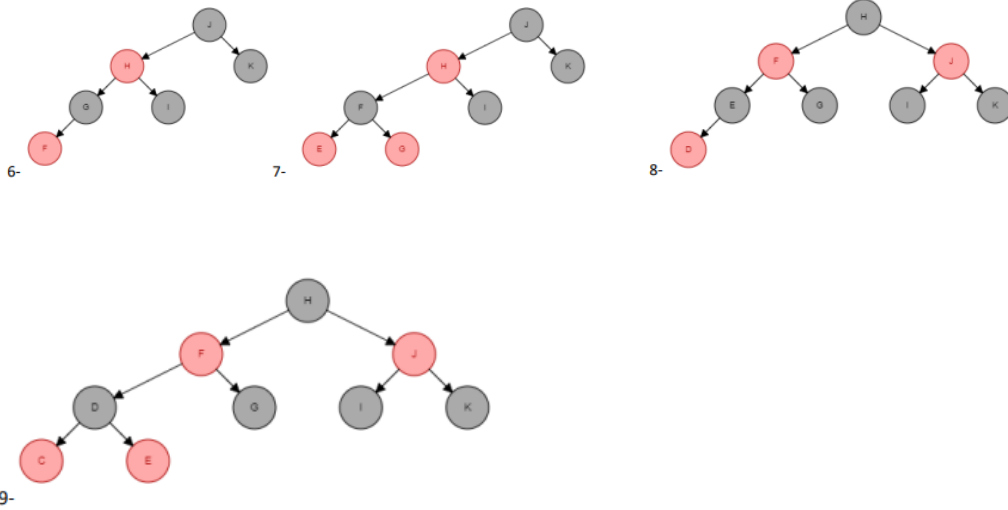
2. Dibuje el paso a paso de insertar las letras desde la A hasta la K, de manera ascendente y descendente, a un red-black BST que está vacío inicialmente.

a. Ascendente.

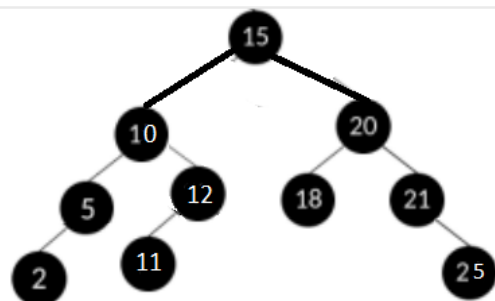
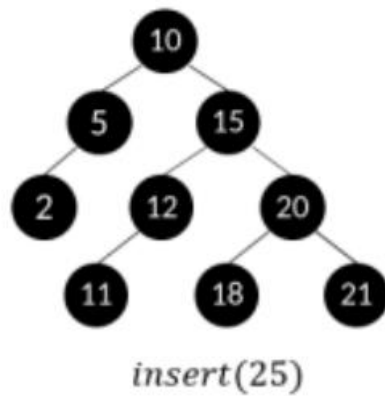
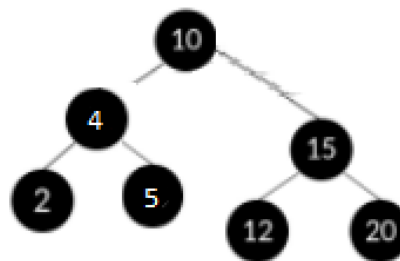
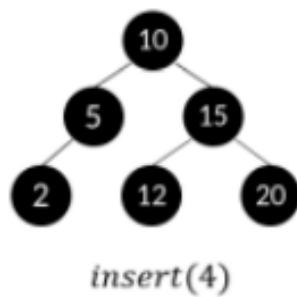


b. Descendente





3. Dibuje el árbol AVL resultante al aplicar las siguientes operaciones:



12.2-1

Suppose that we have numbers between 1 and 1000 in a binary search tree, and we want to search for the number 363. Which of the following sequences could *not* be the sequence of nodes examined?

- a. 2, 252, 401, 398, 330, 344, 397, 363.
- b. 924, 220, 911, 244, 898, 258, 362, 363.
- c. 925, 202, 911, 240, 912, 245, 363.
- d. 2, 399, 387, 219, 266, 382, 381, 278, 363.
- e. 935, 278, 347, 621, 299, 392, 358, 363.

Las secuencias de nodos no examinadas serian la opción c y d, porque la c no es, debido a que 240 viene después de 911, entonces, este debe ser su hijo izquierdo, y cualquier nodo posterior debe ser menor que 911 (padre). Pero el nodo 912 rompe esta secuencia, es mayor que 911 y está en el sub-árbol izquierdo de 911, rompiendo las reglas básicas de los árboles de búsqueda binarios y el e no es, porque esta secuencia es interrumpida por el nodo 299 < 347 y se encuentra en el subárbol derecho de 347, rompiendo las reglas básicas de los árboles de búsqueda binarios

12.2-2

Write recursive versions of TREE-MINIMUM and TREE-MAXIMUM.

```
1 class Node():
2     def __init__(self, val, izquierda = None, derecha = None):
3         self.val = val
4         self.izquierda = izquierda
5         self.derecha = derecha
6
7 def Arbol_Minimo(root):
8     if root is None:
9         return None
10    if root.izquierda is None:
11        return root.val
12    return Arbol_Minimo(root.izquierda)
13
14 def Arbol_Maximo(root):
15     if root is None:
16         return None
17     if root.derecha is None:
18         return root.val
19     return Arbol_Maximo(root.derecha)
20
21
```

12.2-4

Professor Bunyan thinks he has discovered a remarkable property of binary search trees. Suppose that the search for key k in a binary search tree ends up in a leaf. Consider three sets: A , the keys to the left of the search path; B , the keys on the search path; and C , the keys to the right of the search path. Professor Bunyan claims that any three keys $a \in A$, $b \in B$, and $c \in C$ must satisfy $a \leq b \leq c$. Give a smallest possible counterexample to the professor's claim.

Busque 9 en este árbol. Entonces $A = \{7\}$, $B = \{5, 8, 9\}$ y $C = \{\}$. Entonces $7 > 5$.

12.2-5

Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.

Supongamos que el nodo x tiene dos hijos. Por lo tanto, su sucesor es el elemento BST más pequeño. Basado en $x.\text{right}$. Si se deja un hijo en él, no es el elemento más pequeño. entonces no puedes, hay un hijo a la izquierda. Asimismo, el predecesor debe ser el elemento más grande. subárbol izquierdo, por lo que no puede tener un hijo derecho.

12.2-6

Consider a binary search tree T whose keys are distinct. Show that if the right subtree of a node x in T is empty and x has a successor y , then y is the lowest ancestor of x whose left child is also an ancestor of x . (Recall that every node is its own ancestor.)

Primero, especificamos que y debe ser un ancestro de x . Si y no es un ancestro de x , entonces, Sea z el primer ancestro común de x e y . Según la naturaleza del árbol binario de búsqueda, $x < z < y$, por lo que y no puede ser descendiente de x . $y.\text{left}$ debe ser un ancestro de x , porque si no lo fuera, $y.\text{right}$ sería x lo que significa $x > y$. Finalmente, si asumimos que y no es el ancestro más bajo de x , sus hijos a la izquierda también hay un antepasado de x . Denotemos este ancestro más bajo por z . Después debe estar en el subárbol izquierdo de y , lo que significa que zy es descendiente de x .