

Reto 5.

Estructura

de Datos



UNIVERSIDAD
DE GRANADA

*Escuela Técnica Superior de Ingenierías
Informática y de Telecomunicación*

Los Del DGIIM, [losdelDGIIM.github.io](https://github.com/losdelDGIIM)

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

Reto 5.

Estructura de Datos

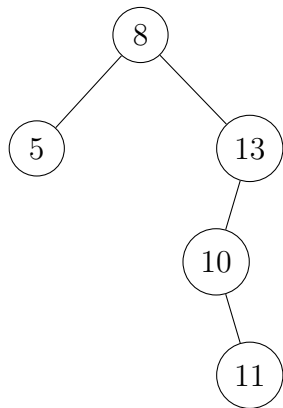
Los Del DGIIM, losdeldgiim.github.io

Arturo Olivares Martos

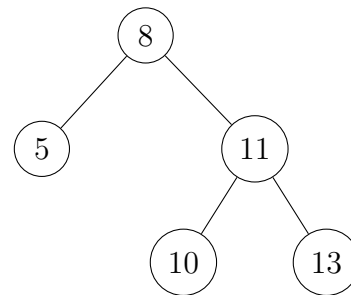
Granada, 2023

Ejercicio 1. En un árbol **AVL** inicialmente vacío insertamos (por este orden) los siguientes elementos: {8, 13, 10, 5, 11, 6, 7}; y después eliminamos el 13. Mostrar la estructura del árbol antes y después de cada operación que requiera una rotación indicando de qué tipo es ésta.

Las primeras inserciones no dan problemas, pero al insertar el 11 se produce un desequilibrio. En el nodo 13, el subárbol izquierdo tiene altura 2, mientras que el subárbol derecho tiene altura 0. Por tanto, hay que realizar una rotación hacia la derecha. Además, es doble, ya que el nodo desequilibrante (el 11) se encuentra en el subárbol derecho del subárbol izquierdo. El resultado de la rotación se ve en la Figura 1.



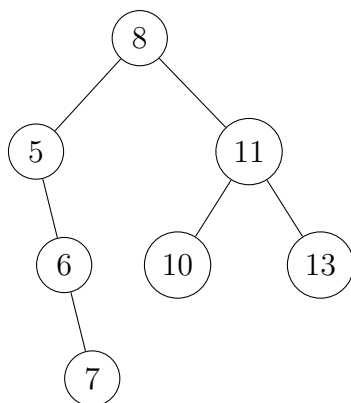
(a) Estructura del AVL antes de la rotación por el desequilibrio por del 11.



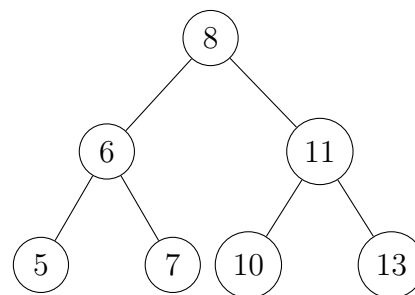
(b) Estructura del AVL después de la rotación por el desequilibrio por del 11.

Figura 1: RDD en el nodo 13 por el desequilibrio que provoca la inserción del 11.

El insertado del 6 tampoco genera problemas, pero al insertar el 7, se desequilibria el 5. En este caso, se trata de una RSI, descrita en la Figura 2.



(a) Estructura del AVL antes de la rotación por el desequilibrio por del 7.



(b) Estructura del AVL después de la rotación por el desequilibrio por del 7.

Figura 2: RSI en el nodo 5 por el desequilibrio que provoca la inserción del 7.

Respecto al borrado del nodo 13, el AVL queda como está representado en la Figura 3. Vemos que ningún nodo queda desequilibrado, por lo que no son necesarias

las rotaciones.

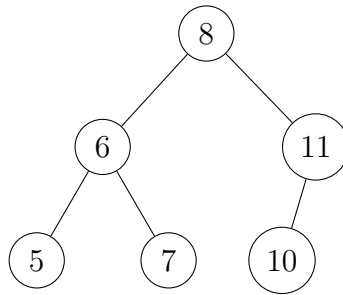


Figura 3: AVL tras el borrado del 13.

Ejercicio 2. Resuelva los siguientes ejercicios:

1. Insertar los enteros $\{8,3,10,1,6,7,9,2,11\}$ en un **APO**. Obtener los árboles resultantes de aplicar el borrado dos veces.

Tenemos que el APO tras insertar todos los elementos es el representado en la Figura 4:

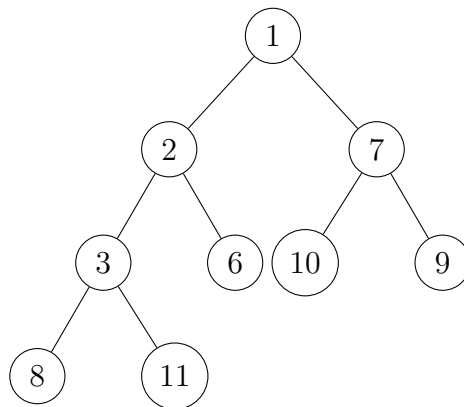
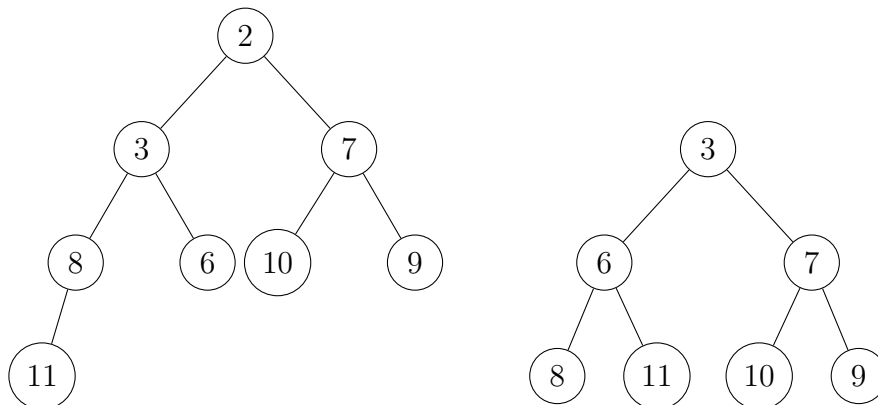
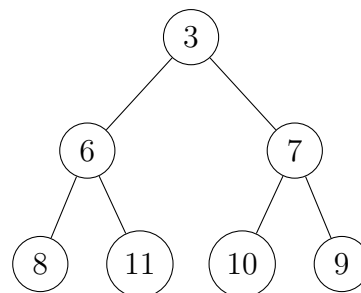


Figura 4: APO tras la inserción de todos los elementos.

Tras ejecutar el primer borrado, el APO queda representado en la Figura 5a, mientras que tras el segundo borrado se muestra en la Figura 5b:



(a) APO tras el primer borrado.



(b) APO tras el segundo borrado.

Figura 5: Dos borrados del APO de la Figura 4.

| Posición | Clave | Status |
|----------|-------|--------|
| 0 | 11 | O |
| 1 | | L |
| 2 | 13 | O |
| 3 | 62 | O |
| 4 | 59 | O |
| 5 | 5 | O |
| 6 | 17 | O |
| 7 | 7 | O |
| 8 | 38 | O |
| 9 | 24 | O |
| 10 | 10 | O |

Tabla 1: Tabla Hash cerrada tras las primeras 10 inserciones.

2. Insertar las claves $\{5, 13, 17, 38, 7, 59, 24, 62, 10, 11\}$ en una **Tabla Hash cerrada** de tamaño 11. A continuación borrar el 11 y el 38 y finalmente insertar el valor 49. Resolver las colisiones usando **hashing doble**.

Como función hash, usamos $h(k) = k \% 11$. En el caso de que se produzcan colisiones, la función de rehashing que usamos es:

$$h_i(k) = (h_{i-1}(k) + h_0(k)) \% 11 \quad \forall i = 2, 3, \dots \quad \begin{cases} h_0(k) = 1 + k \% 9 \\ h_1(k) = h(k) \end{cases}$$

Entonces, tenemos que:

$$\begin{aligned} h(5) &= 5 \\ h(13) &= 2 \\ h(17) &= 6 \\ h(38) &= 5 & h_2(38) &= 5 + h_0(38) = 5 + 3 = 8 \\ h(7) &= 7 \\ h(59) &= 4 \\ h(24) &= 2 & h_2(24) &= 2 + h_0(24) = 2 + 7 = 9 \\ h(62) &= 7 & h_2(62) &= (7 + 9) \% 11 = 5 & h_3(62) &= (5 + 9) \% 11 = 3 \\ h(10) &= 10 \\ h(11) &= 0 \end{aligned}$$

Tenemos que la tabla tras las primeras 10 inserciones queda conforme a la Tabla 1.

Tras el borrado del 11 y el 38, tan solo es necesario buscarlas y marcarlas como borradas. La tabla quedará como la Tabla 2.

Insertamos ahora el 49. Veamos su posición:

$$\begin{aligned} h(49) &= 5 & h_2(49) &= 10 & h_3(49) &= (10 + 5) \% 11 = 4 & h_4(49) &= 4 + 5 = 9 \\ & & h_5(49) &= (9 + 5) \% 11 = 3 & h_6(49) &= 3 + 5 = 8 \end{aligned}$$

Por tanto, insertamos el 49 en la posición 8, cuyo estado es B y por tanto, en términos de inserción, disponemos de un hueco. La tabla quedará como la Tabla 3.

| Posición | Clave | Status |
|----------|-------|--------|
| 0 | | B |
| 1 | | L |
| 2 | 13 | O |
| 3 | 62 | O |
| 4 | 59 | O |
| 5 | 5 | O |
| 6 | 17 | O |
| 7 | 7 | O |
| 8 | | B |
| 9 | 24 | O |
| 10 | 10 | O |

Tabla 2: Tabla Hash cerrada tras el borrado del 11 y el 38.

| Posición | Clave | Status |
|----------|-------|--------|
| 0 | | B |
| 1 | | L |
| 2 | 13 | O |
| 3 | 62 | O |
| 4 | 59 | O |
| 5 | 5 | O |
| 6 | 17 | O |
| 7 | 7 | O |
| 8 | 49 | O |
| 9 | 24 | O |
| 10 | 10 | O |

Tabla 3: Tabla Hash cerrada final tras la inserción del 49.