

Árboles Isomorfos

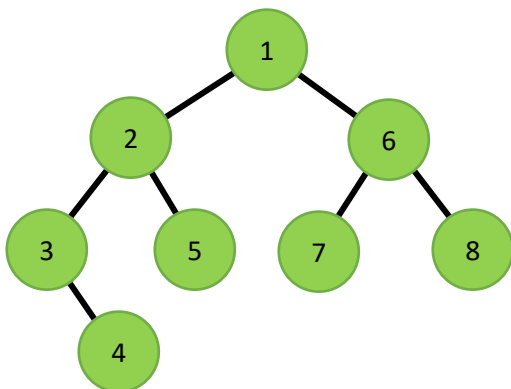
NOTA: Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

Árboles isomorfos

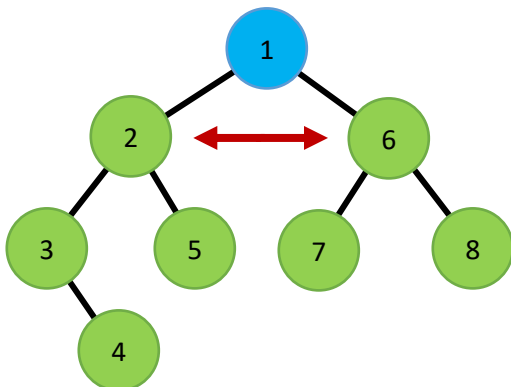
Dos árboles son **isomorfos** si, luego de realizadas permutaciones en el orden de los hijos de cualquiera de sus subárboles, son iguales. De manera intuitiva es poder “llegar” desde un árbol a otro realizando permutaciones de sus hijos. En el caso del examen, **los árboles a considerar serán siempre binarios**, aunque la definición de árboles isomorfos no se restringe solo a los de este tipo.

Por ejemplo, los siguientes árboles (A y B) son isomorfos. Considere las **permutaciones** siguientes realizadas sobre el árbol A. Note que en los ejemplos los valores de los nodos son números enteros, pero podrían ser de cualquier tipo.

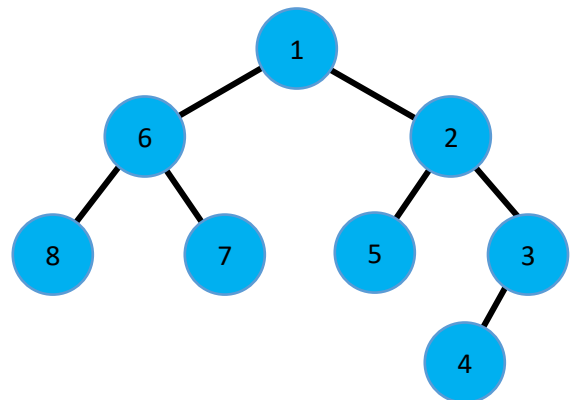
Árbol A



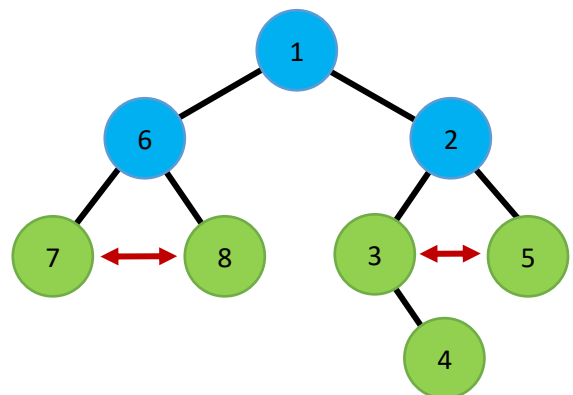
Permutación 1 (Árbol inicial A)



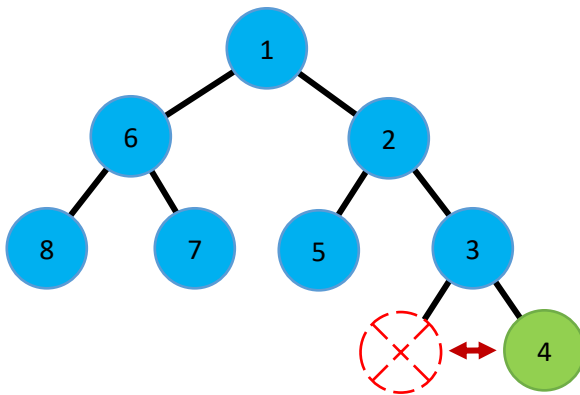
Árbol B



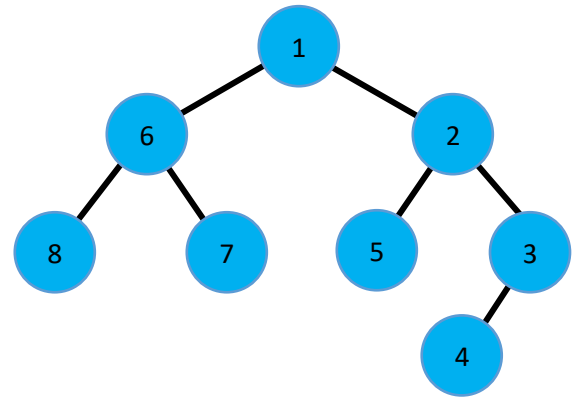
Permutaciones 2 y 3



Permutación 4



Árbol final (Árbol B)

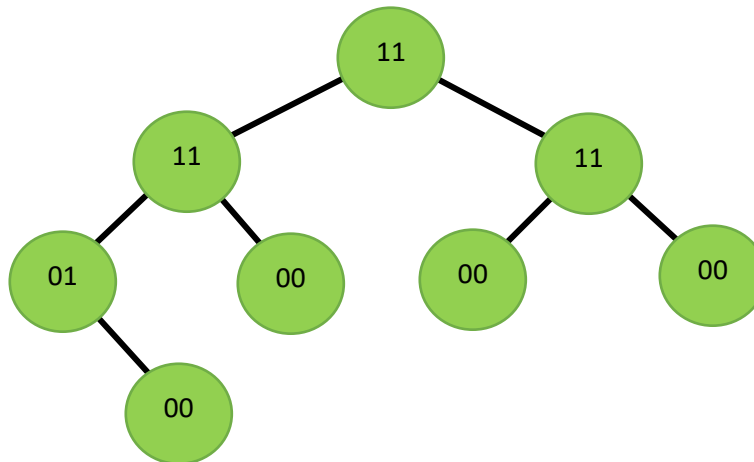


Nota: En la permutación 4, el nodo con valor 3 solo tiene un nodo hijo asignado a la derecha por lo que al realizar la transformación pasa a ser hijo izquierdo del mismo nodo.

Comparación de árboles

Adicionalmente, cada nodo puede ser etiquetado con dos números, los cuales pueden ser 0 o 1. Esta etiqueta representa la presencia o no de un hijo. Si un nodo tiene ambos hijos, entonces su etiqueta será "11". Si un nodo tiene hijo izquierdo, pero no tiene hijo derecho, su etiqueta será "10". Si tiene el hijo derecho, pero no el izquierdo, la etiqueta será "01" y finalmente, si no tiene hijos, la etiqueta será "00". En la siguiente figura se muestran las etiquetas del árbol A.

Árbol A etiquetado



Empleando estas etiquetas es posible obtener una representación de un árbol binario como una cadena formada por ceros y unos. Se concatenan las etiquetas según el recorrido en **preorden** del árbol. De esta manera es posible establecer un orden entre dos árboles cualesquiera (orden lexicográfico). Para determinar si una cadena es menor que otra basta con comparar carácter a carácter siendo menor la que tenga el menor valor en el primer número en que se diferencien.

Si ambas cadenas binarias fuesen iguales entonces se compara nuevamente, pero considerando los valores de los nodos. **Todos los tipos de los valores almacenados implementan la interfaz IComparable.** Si los valores son iguales entonces se asume que no hay ninguno menor que otro. **Puede asumir que los árboles a comparar siempre serán isomorfos.**

Ejemplo:

Representación del árbol A: **11 11 01** 00 00 11 00 00

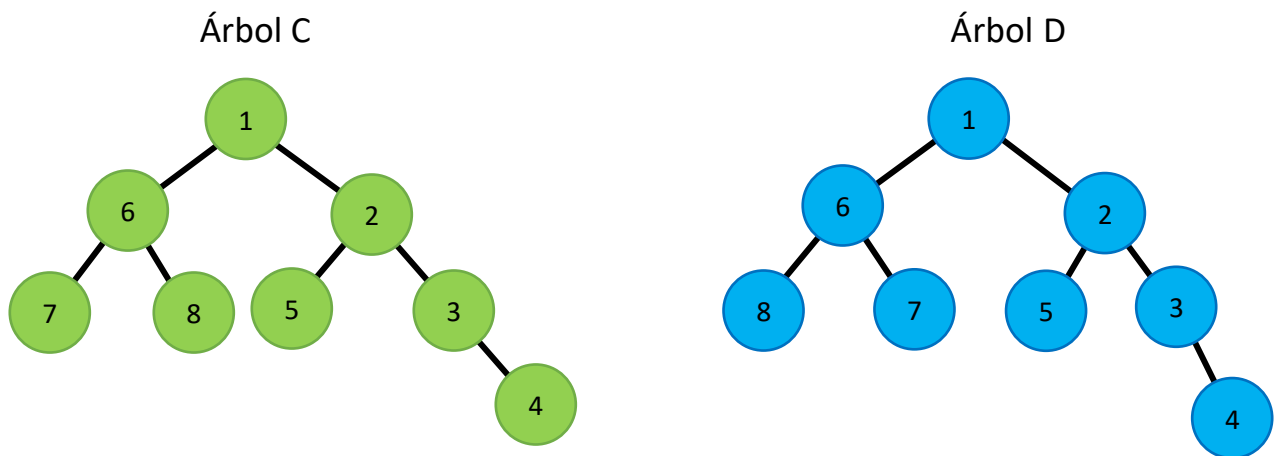
Representación del árbol B: **11 11 00** 00 11 00 10 00

En este caso, el primer número en que se diferencian es en el que ocupa la posición 6, el árbol B tiene valor 0 y el árbol A tiene valor 1, y por tanto el árbol B es menor que el árbol A.

Familia de árboles isomorfos

Por último, se define como una familia de árboles isomorfos, a un conjunto de árboles en el cual todos son isomorfos entre sí y, además, no existe ningún otro árbol que sea isomorfo con ellos que no pertenezca al conjunto.

El menor árbol de una familia de árboles isomorfos es el que cumple que siempre es menor al compararlo con cualquier otro perteneciente al mismo conjunto. Se aplica el mismo criterio de comparación visto anteriormente (etiquetas y preorden). En el siguiente ejemplo, se muestran los árboles C y D, que son los árboles de la familia de isomorfos a la cual pertenecen A y B, y que tienen la menor cadena de etiquetas lexicográficamente. Sin embargo, se considera a C como el menor árbol de la familia, porque la cadena de valores de los nodos que representa su recorrido en preorden es menor que la de D.



En este ejemplo, los preorden por etiquetas de cada árbol son:

Árbol C: 11 11 00 00 11 00 01 00

Árbol D: 11 11 00 00 11 00 01 00

Como son iguales, se procede a analizar los recorridos en preorden de cada árbol. En este caso, serían:

Árbol C: **1 6 7** 8 2 5 3 4

Árbol D: **1 6 8** 7 2 5 3 4

Por tanto, como el preorden del árbol C es menor lexicográficamente que el preorden del árbol D, entonces el árbol C es el menor de la familia de isomorfos.

Su tarea consiste en implementar tres métodos en la clase **ArbolesIsomorfos** que se encuentra en la solución proporcionada junto a este documento:

1. **SonIsomorfos**: Dados dos árboles binarios, devuelve true si son isomorfos entre sí, false en caso contrario.
2. **EsMenor**: Dado dos árboles binarios isomorfos, devuelve true si el primero es menor que el segundo, false en cualquier otro caso.
3. **ObtenerMenorEnFamilia**: Dado un árbol binario isomorfo, devuelve el menor árbol de su familia.

NOTA: Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará y su prueba quedará invalidada.

NOTA: Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y dé buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.