



RETO 4 ARBOLES

Estructura de datos

Jose Antonio Padial Molina, Jose Antonio Martín Alarcón, Elena Ortiz Moreno
josepadial@correo.ugr.es

Contenido

Enunciado.....	2
Primera idea	2
Segunda idea	3
Optimización de la idea inicial.....	4
Ejemplos de la optimización de la idea inicial con árboles nivel 3.....	5
Optimización de la idea inicial para árboles nivel 4 y posterior.....	6

Enunciado

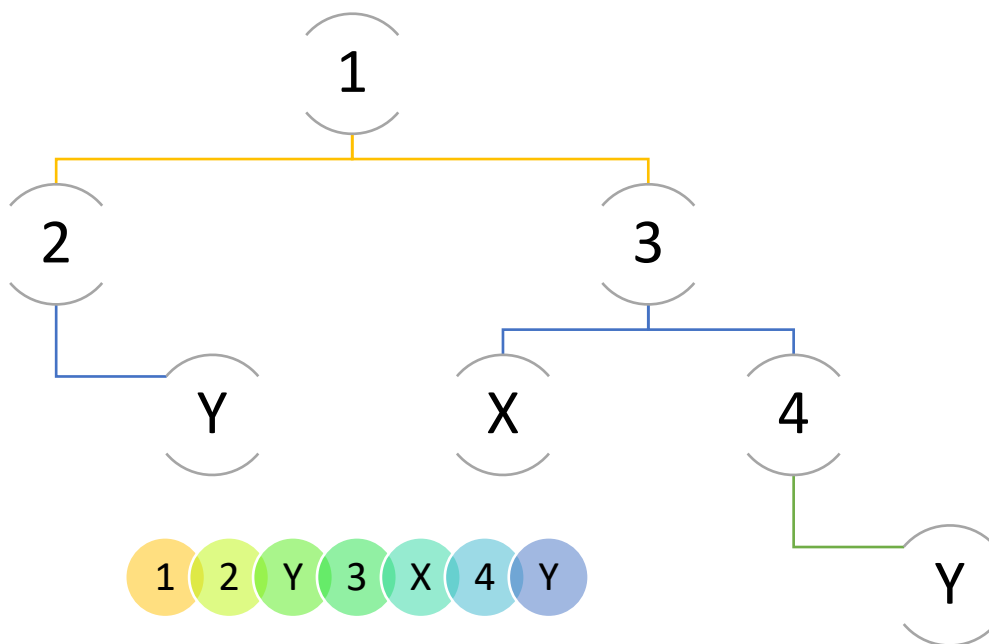
Dar un procedimiento para guardar un árbol binario en disco de forma que se recupere la estructura jerárquica de forma unívoca usando el mínimo número de centinelas que veáis posible. Es una competición para conseguir dar la mejor solución, que será la que use un menor número de datos para el proceso de salvado a disco del árbol binario.

Primera idea

Vamos a tener dos tipos de centinelas:

1. **X** -> para los nodos que solo tienen un hijo
2. **Y** -> para los nodos que no tiene hijos

Vamos a hacer un ejemplo de un árbol con recorrido en preorden:

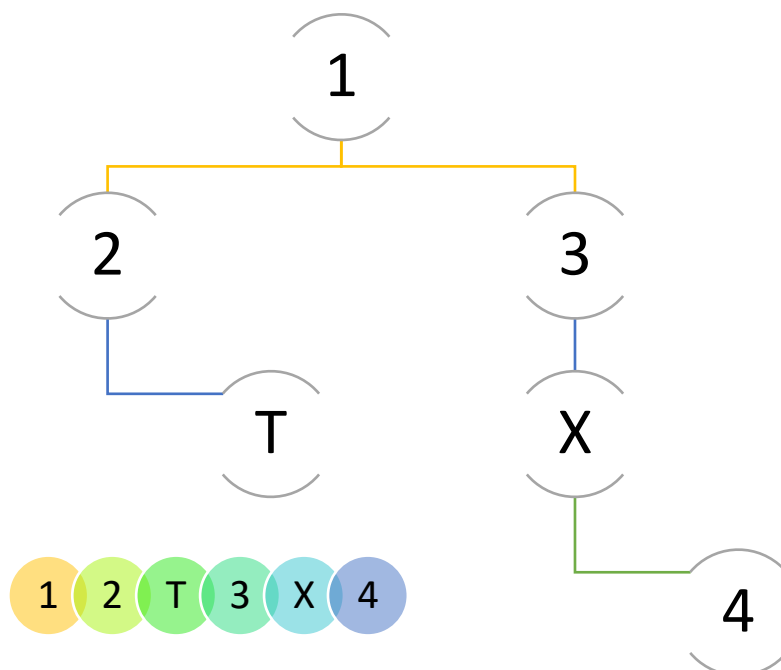
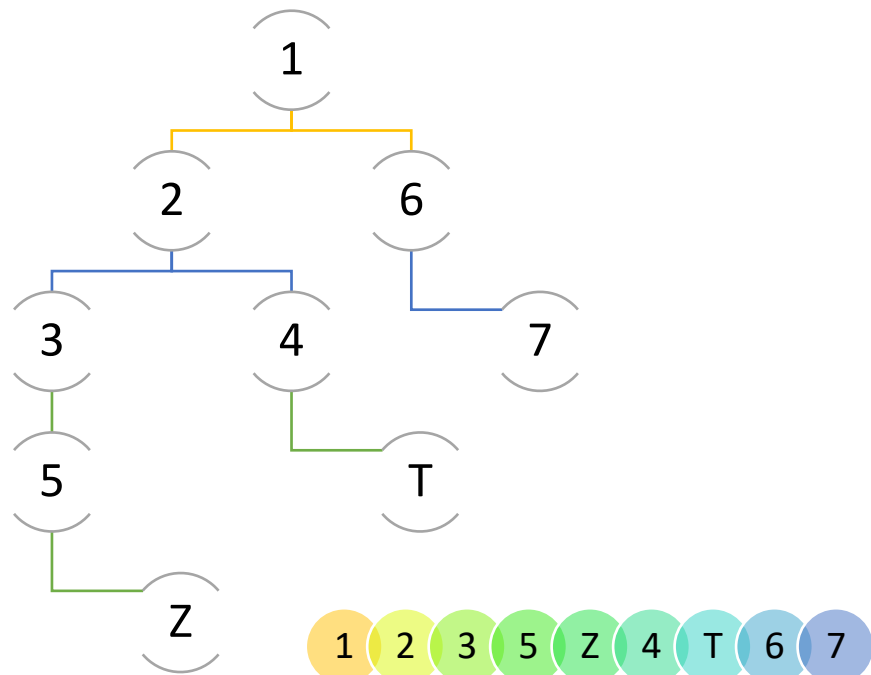


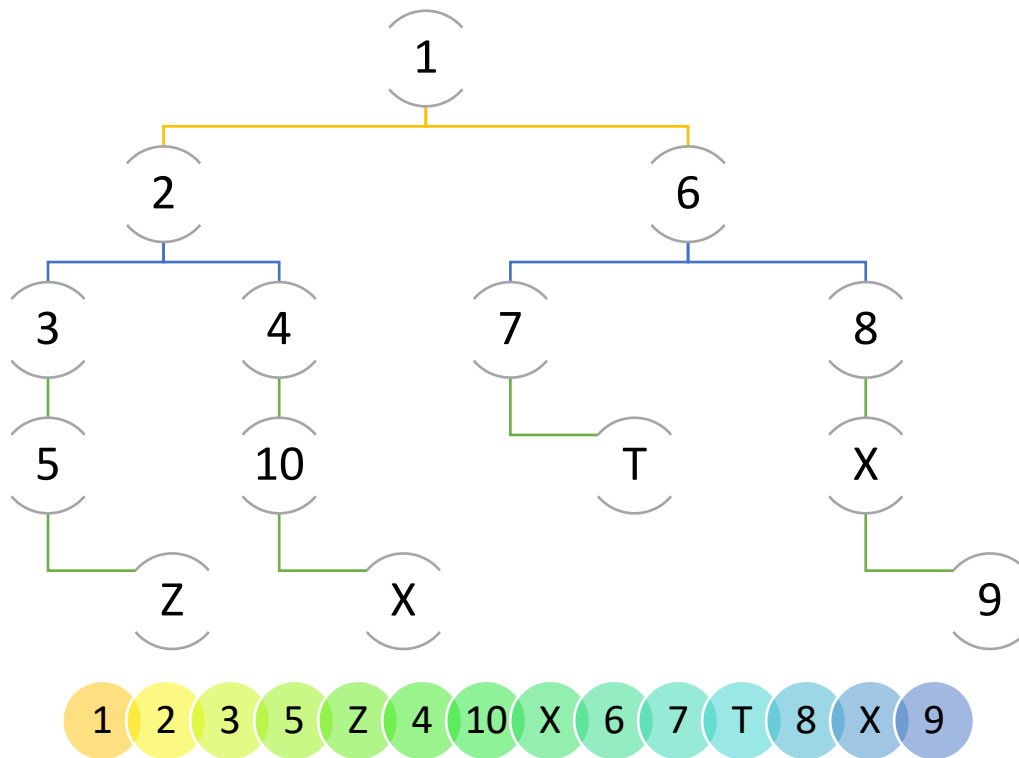
Segunda idea

Esto es una segunda idea posible, la cual la indicamos, pero no la damos como solución final. Como solución final será una [optimización de la idea inicial](#). Como en la idea inicial se van a recorrer también en preorden los árboles. Donde se van a representar los nodos de la siguiente forma:

1. **X** -> Si no hay hijo a la derecha o a la izquierda.
2. **T** -> Si no hay hijo ni a la izquierda ni a la derecha.
3. **Z** -> La unión de las condiciones **X** e **Y**.

Para ellos vamos a ver algunos ejemplos:





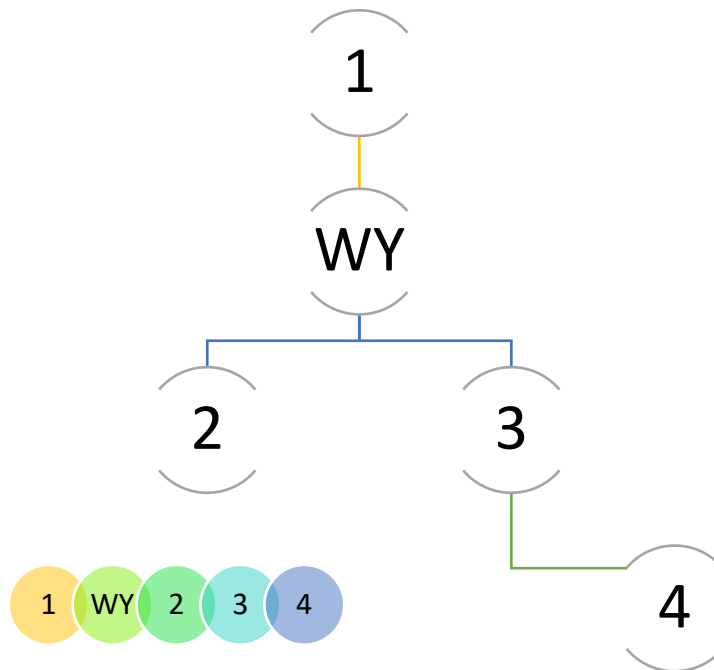
Optimización de la idea inicial

Vamos a cambiar la función de los centinelas. En la idea inicial los hemos usado de topes, dono lo que hacían era indicar el fin del árbol por donde se hallaba él. La idea actual es que los centinelas den información sobre acerca de la forma del árbol. Para representar los árboles que siguen esta idea vamos a usar la siguiente nomenclatura:

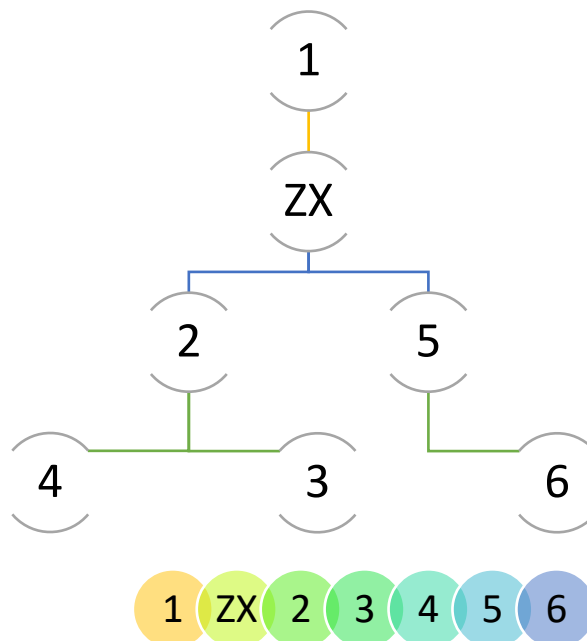
1. **O** -> Nodo sin hijos.
2. **W** -> Nodo con un hijo, el cual no tiene más hijos.
3. **X** -> Nodo con un hijo, el cual solo tiene un hijo izquierdo.
4. **Y** -> Nodo con un hijo, el cual solo tiene un hijo derecho.
5. **Z** -> Nodo con un hijo, el cual tiene dos hijos.

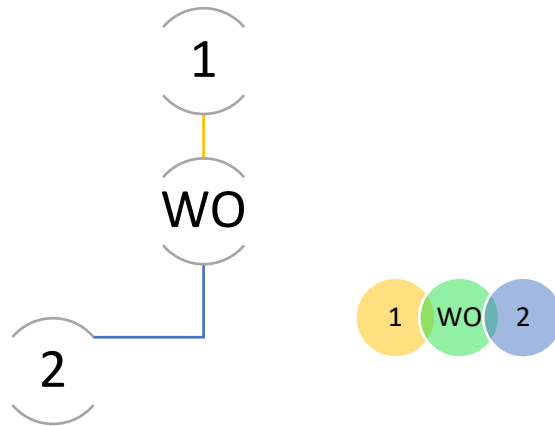
Ejemplos de la optimización de la idea inicial con árboles nivel 3

Aplicando la nueva nomenclatura al árbol de la [primera idea](#), donde vamos a reducir de tres centinelas a uno solo:



Vamos a realizar unos cuantos ejemplos más sobre árboles y como quedarían en el disco:





Optimización de la idea inicial para árboles nivel 4 y posterior

La resolución ideal será ampliar la nomenclatura para con un solo centinela conocer la forma completa del árbol. Y así ampliar la nomenclatura para alturas mayores, para que siempre se pueda generar el árbol con un solo centinela.