

Raíces y raíces

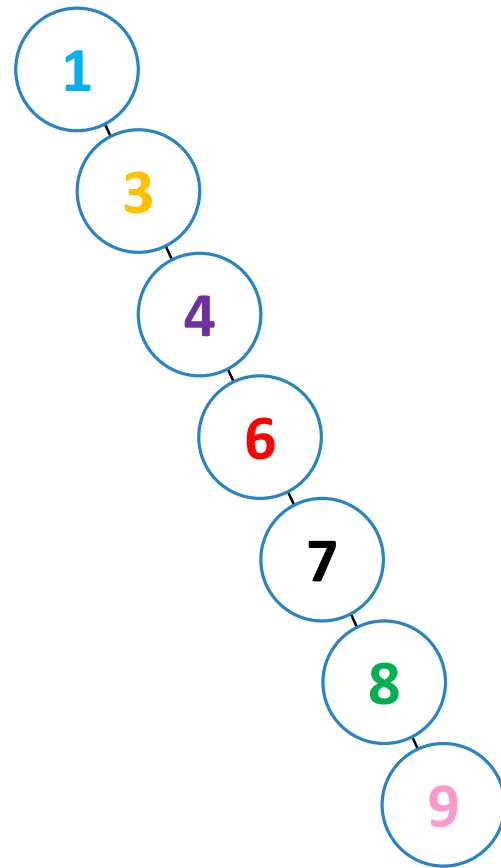
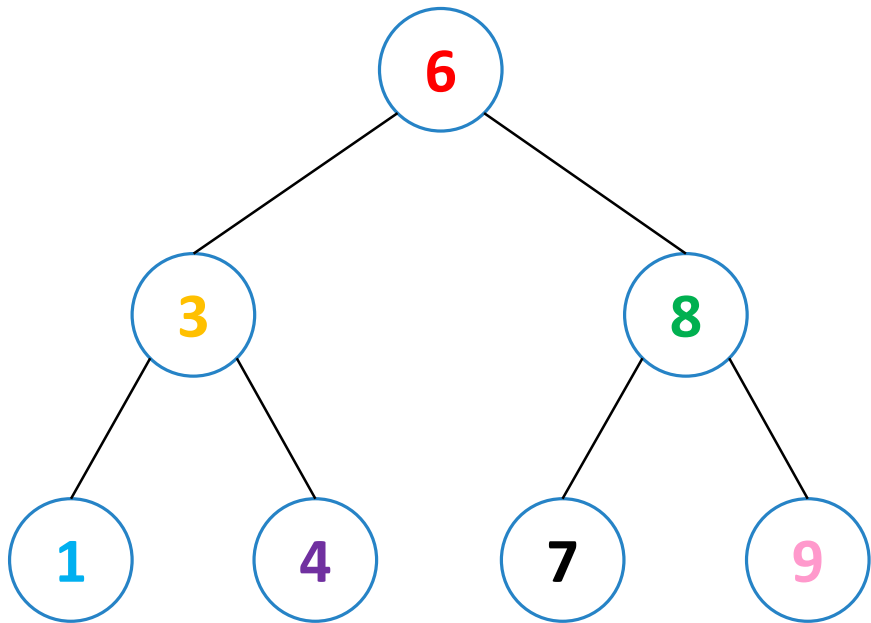


¿Hay algunas raíces más convenientes que otras?

¿Qué pasa con el árbol si no queda un dato conveniente como raíz?

¿Cómo varía la complejidad de las operaciones?

Mismos datos, distinto árboles



¡Ya no podemos garantizar altura $O(\log n)$!

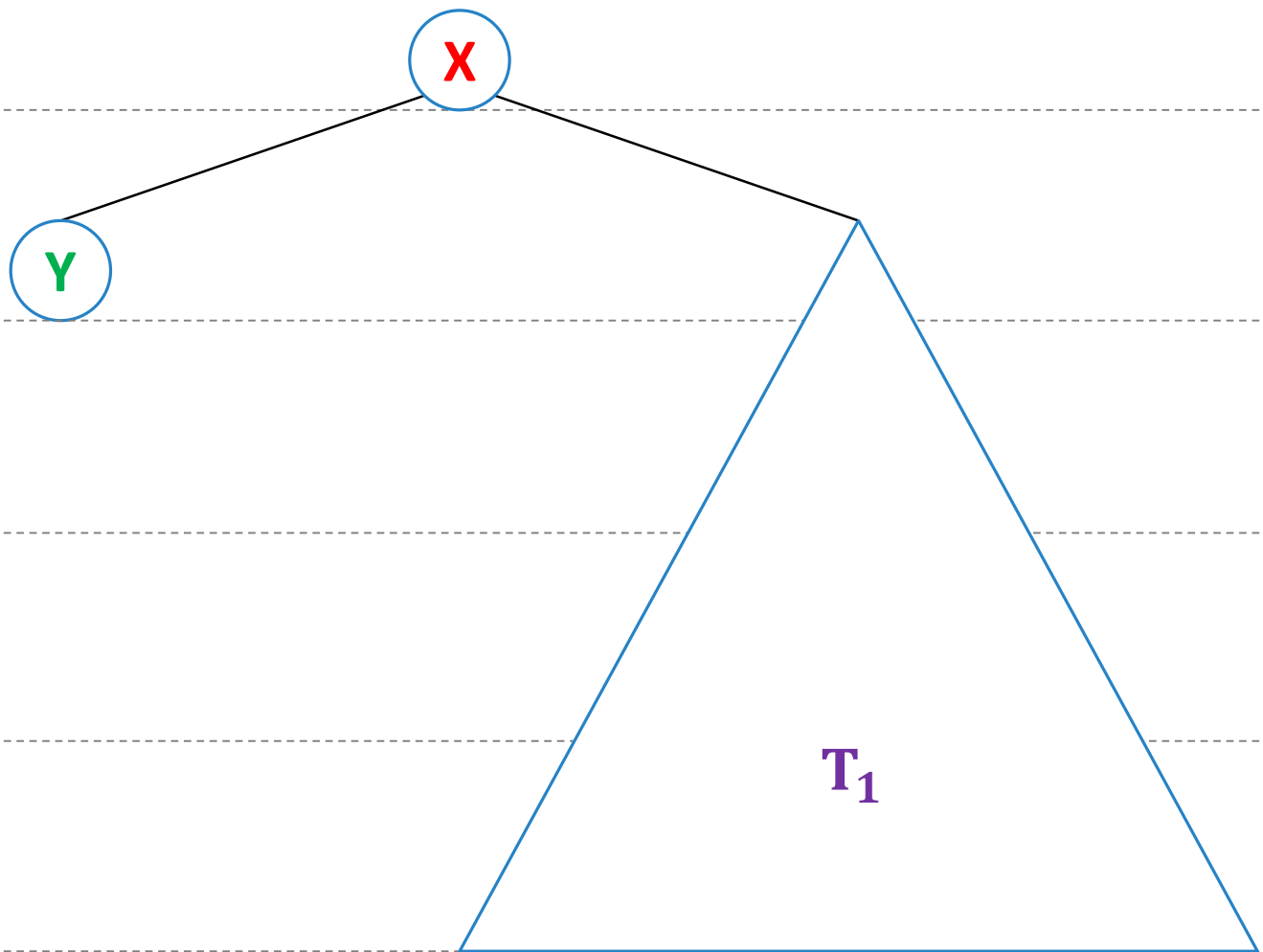
Queremos asegurarnos de que el árbol esté **balanceado**



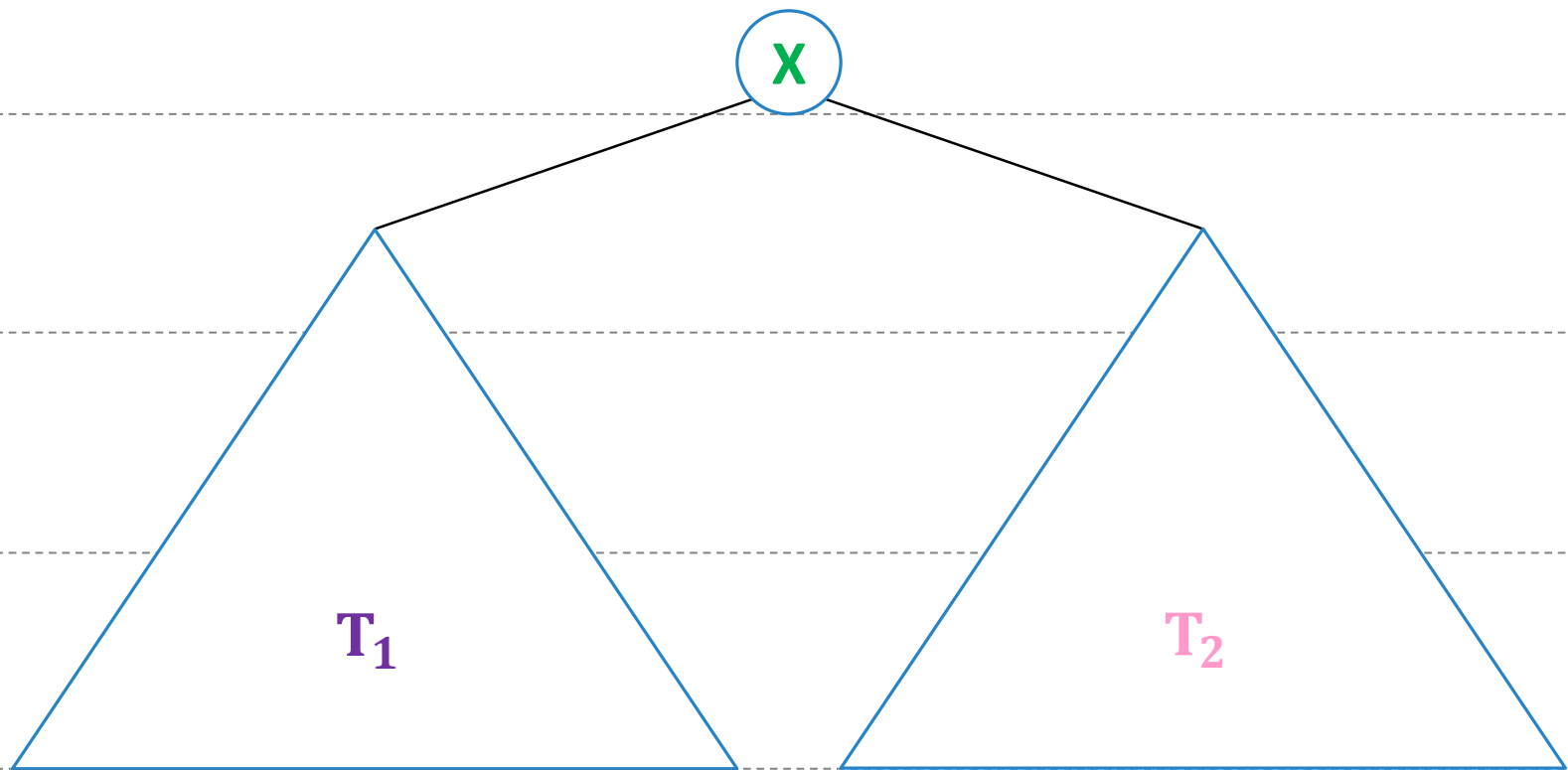
¿Cómo podríamos definir esta noción?

Nos interesa que se pueda cumplir recursivamente

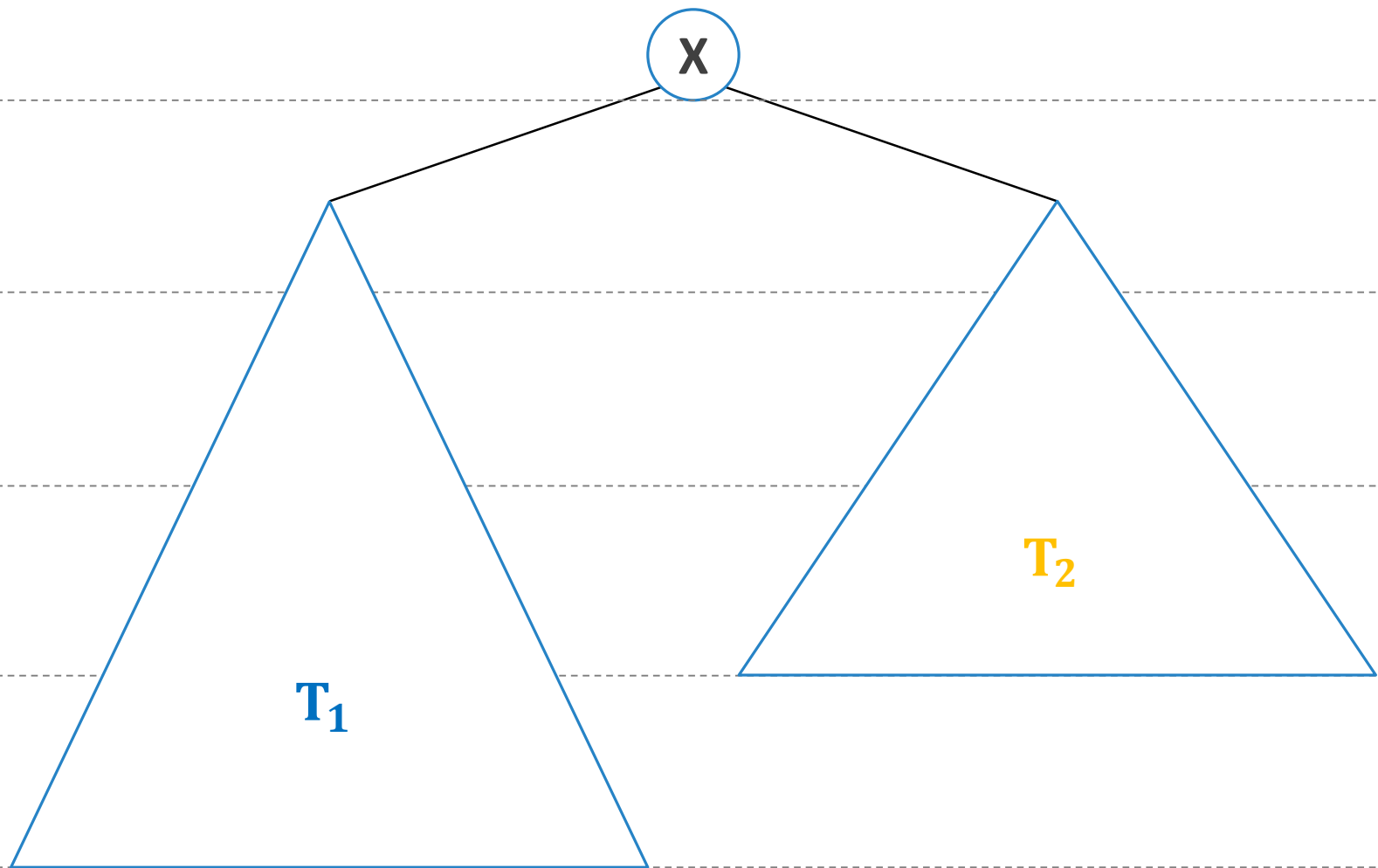
¿Está balanceado?



¿Está balanceado?



¿Está balanceado?



Balance AVL

Diremos que un ABB está **AVL-balanceado** si:

- Las alturas de sus hijos no difieren en más que 1 entre ellas
- Cada hijo a su vez está **AVL-balanceado**

Un ABB que cumple esta propiedad se llama **árbol AVL**

Operaciones en árboles AVL

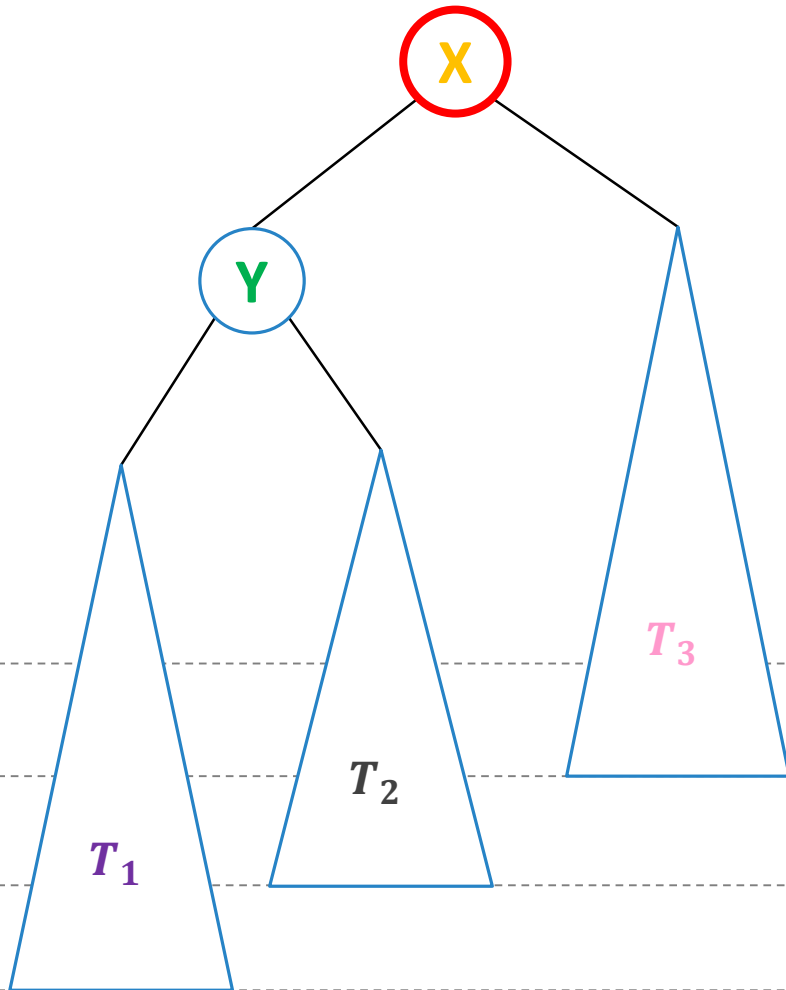


Al insertar o eliminar un nodo, es posible desbalancear el árbol

¿Cómo garantizamos el **balance** del árbol luego de cada operación?

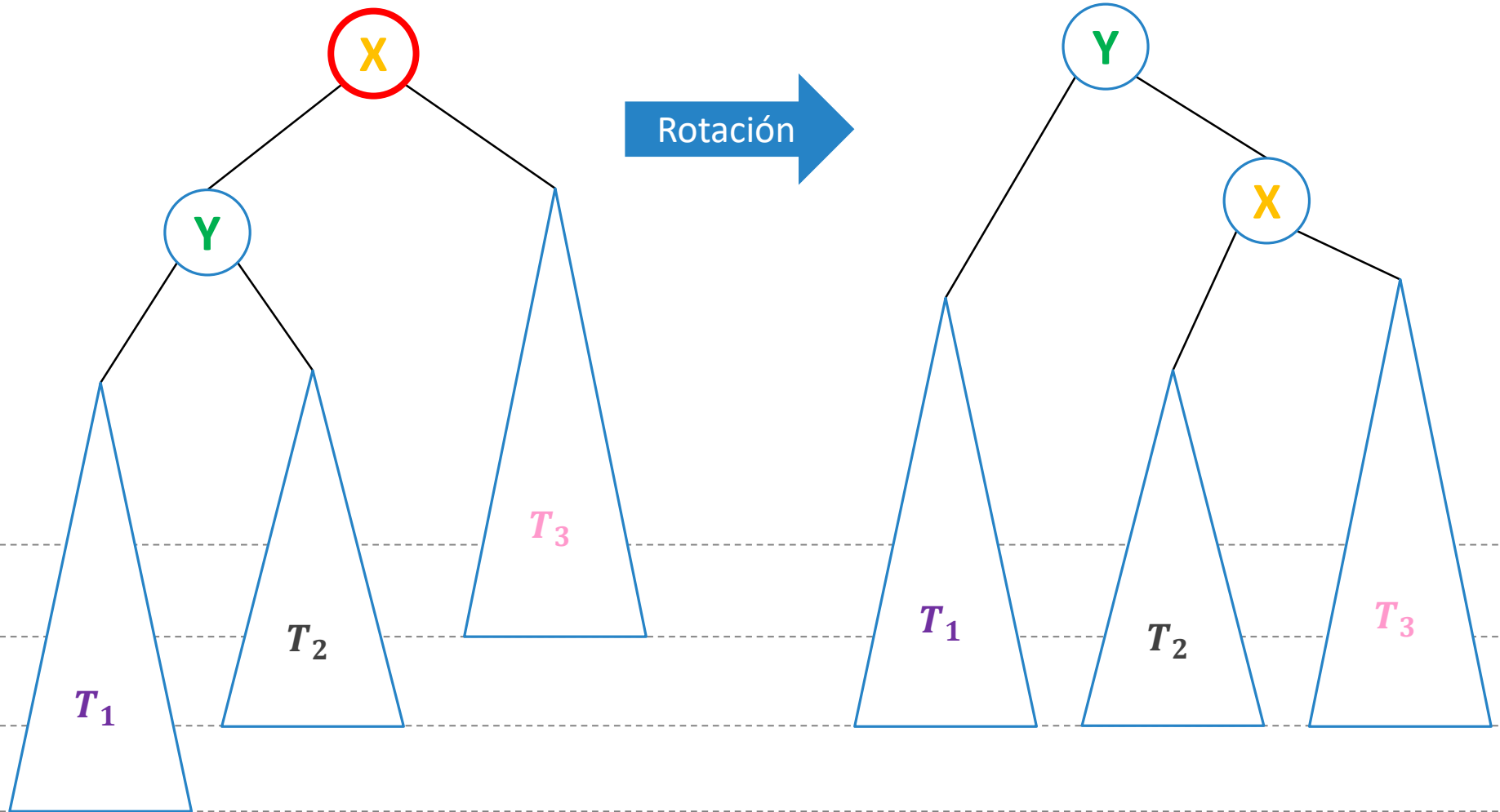
Nos interesa conservar todas las propiedades de los ABB

Luego de hacer una inserción en T_1

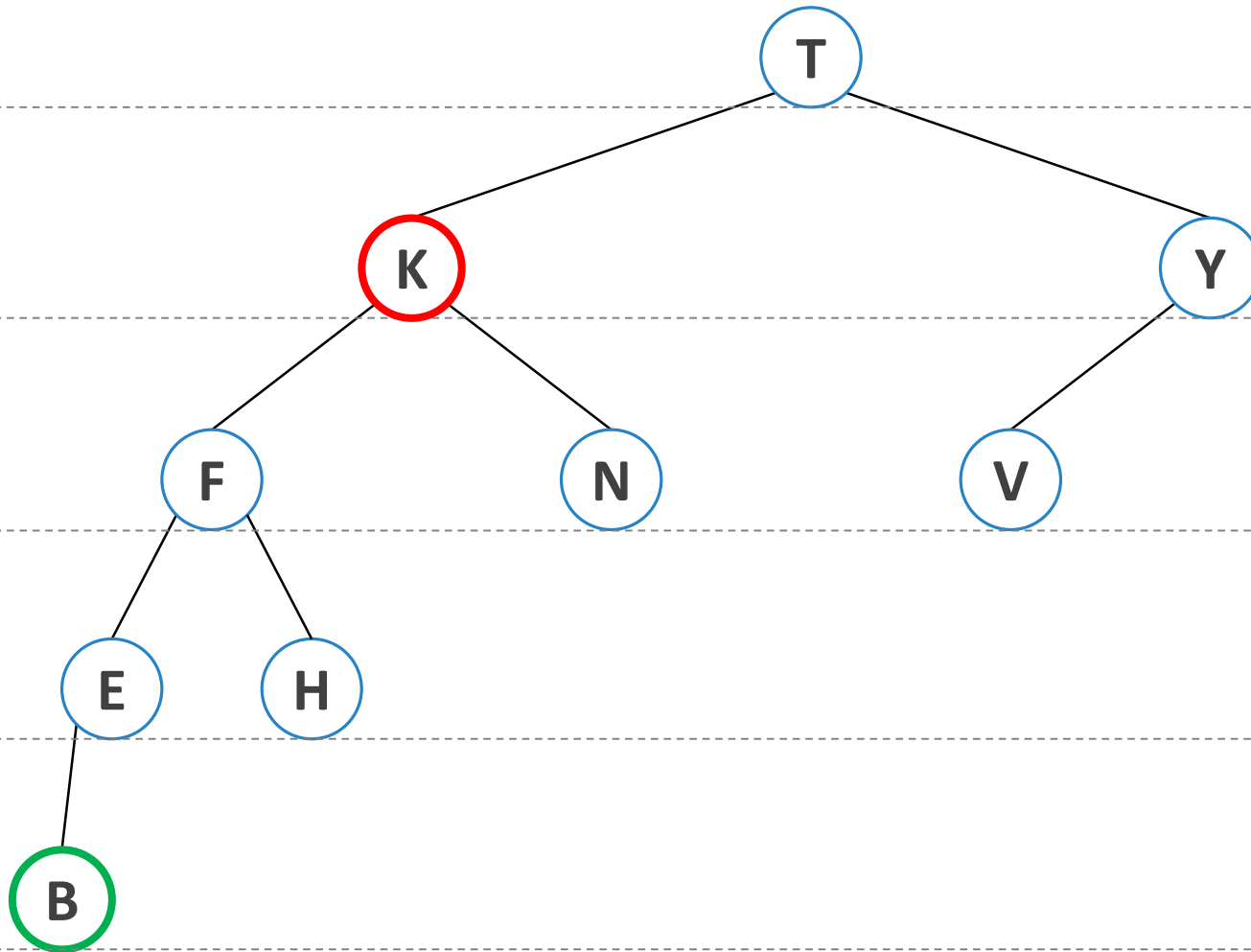


¿Cómo (re)balancear el
árbol con raíz **X**?

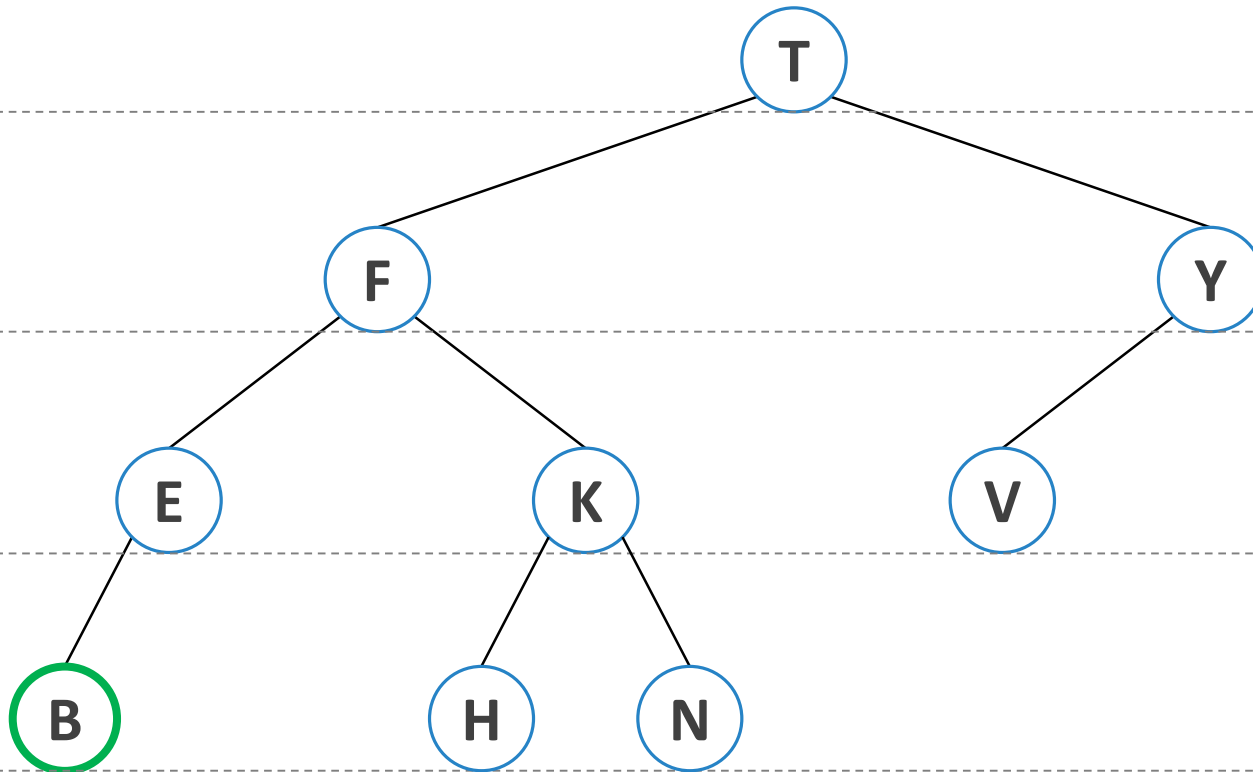
Rotación a la derecha en torno a X - Y



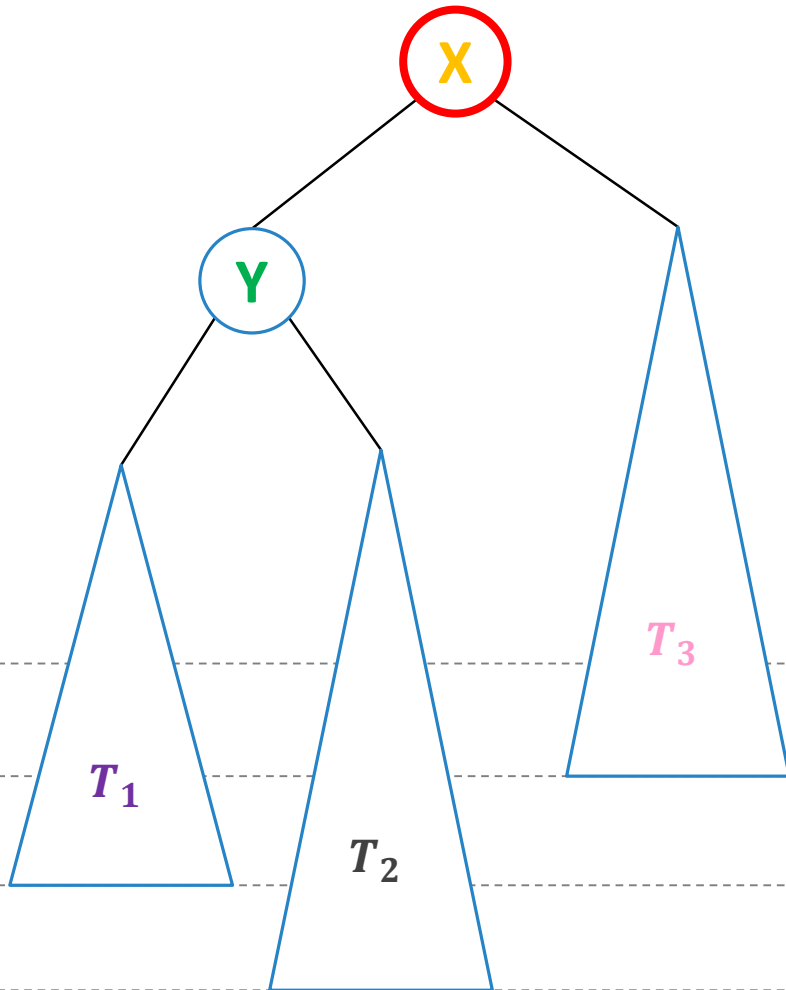
Luego de insertar B



Rotación a la derecha en torno a $K-F$

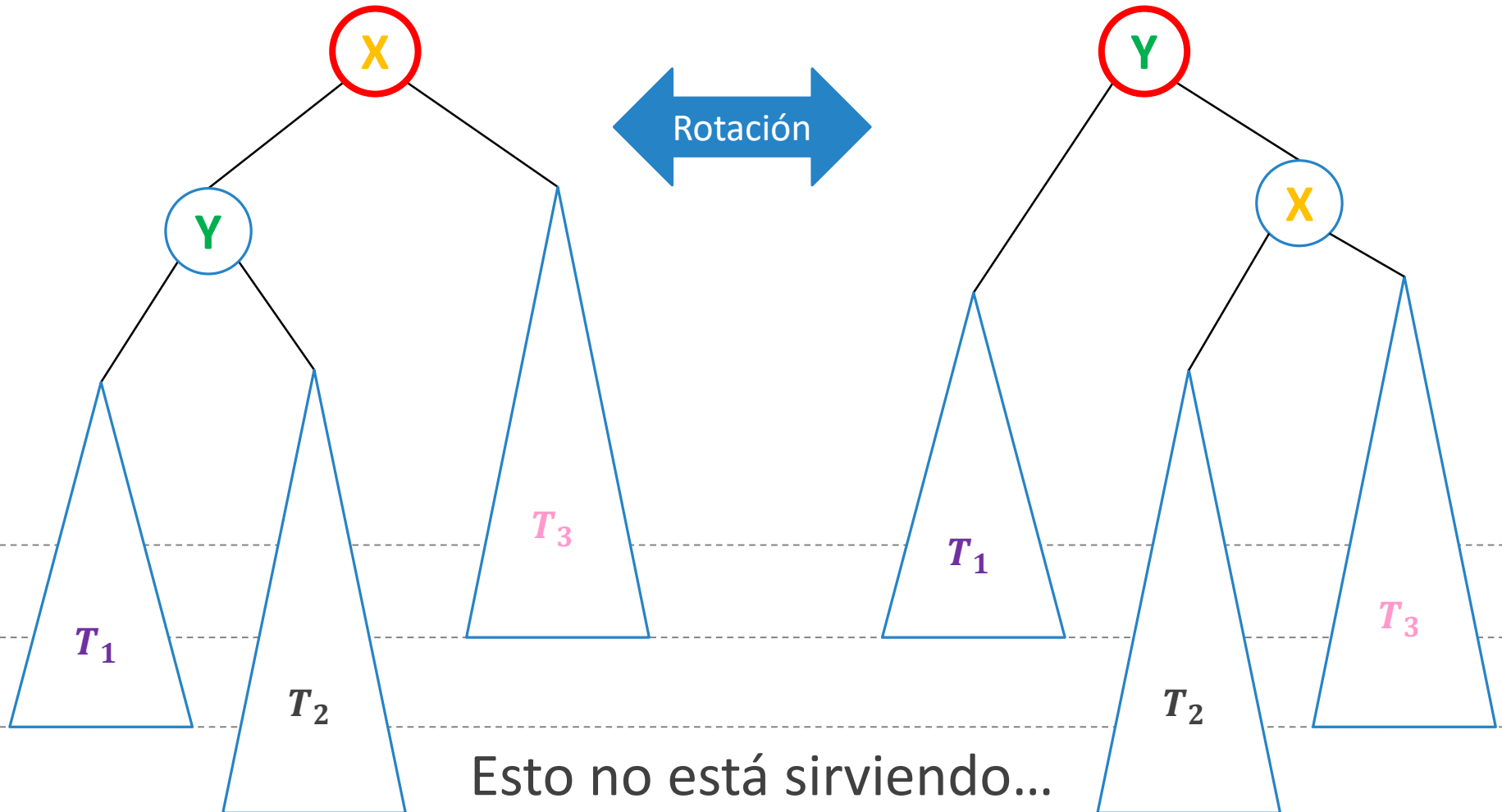


Luego de inserción en T_2

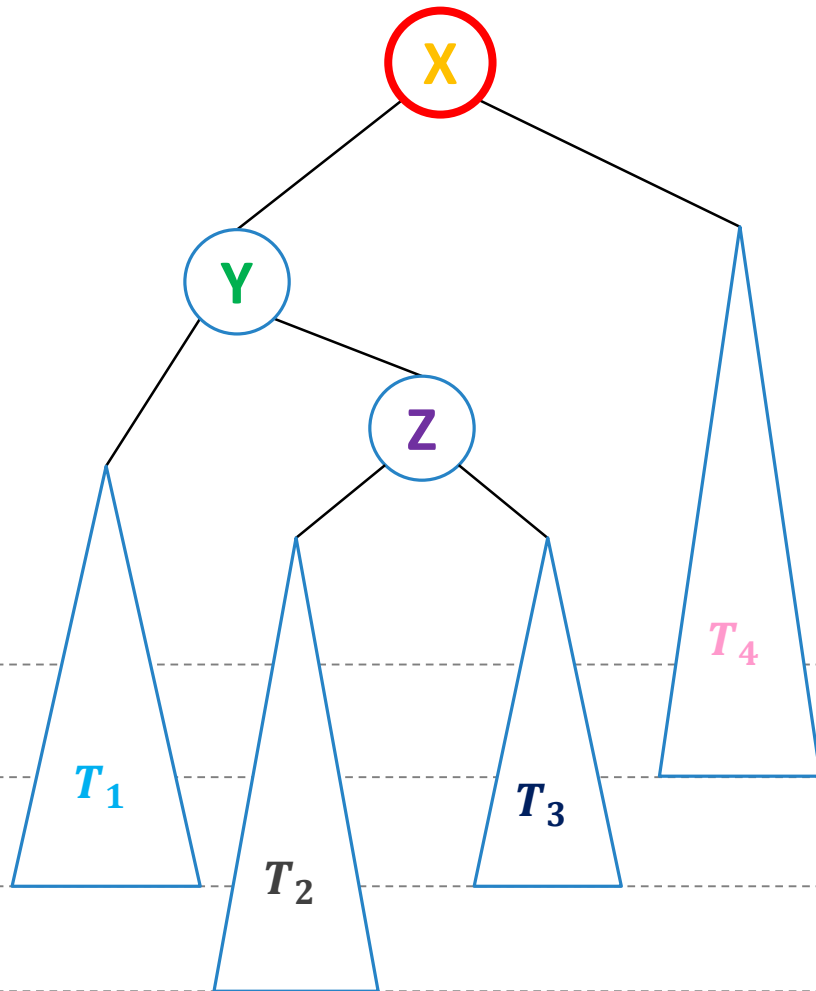


¿Cómo (re)balancear el
árbol con raíz **X**?

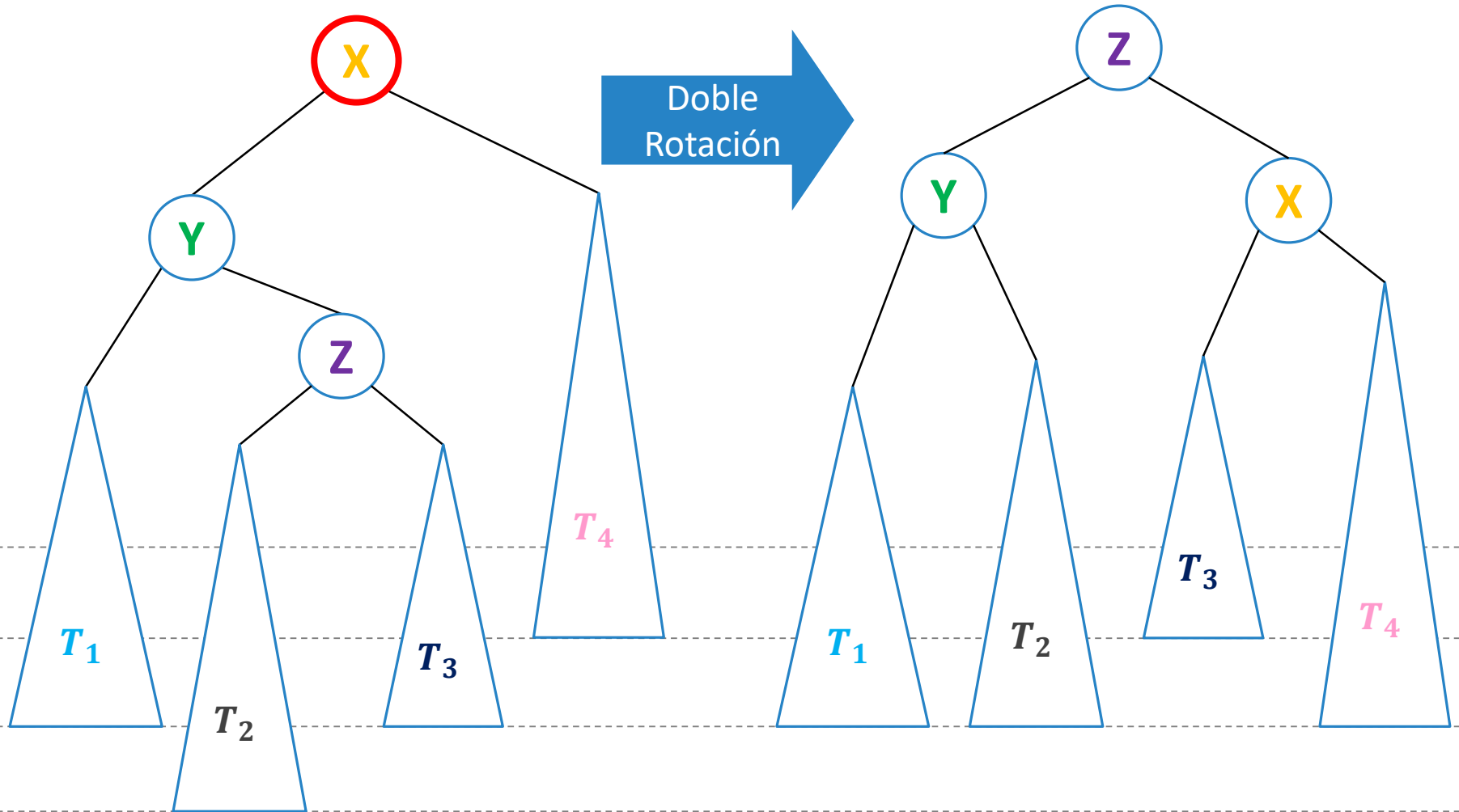
¿Rotación a la derecha en torno a X - Y ?



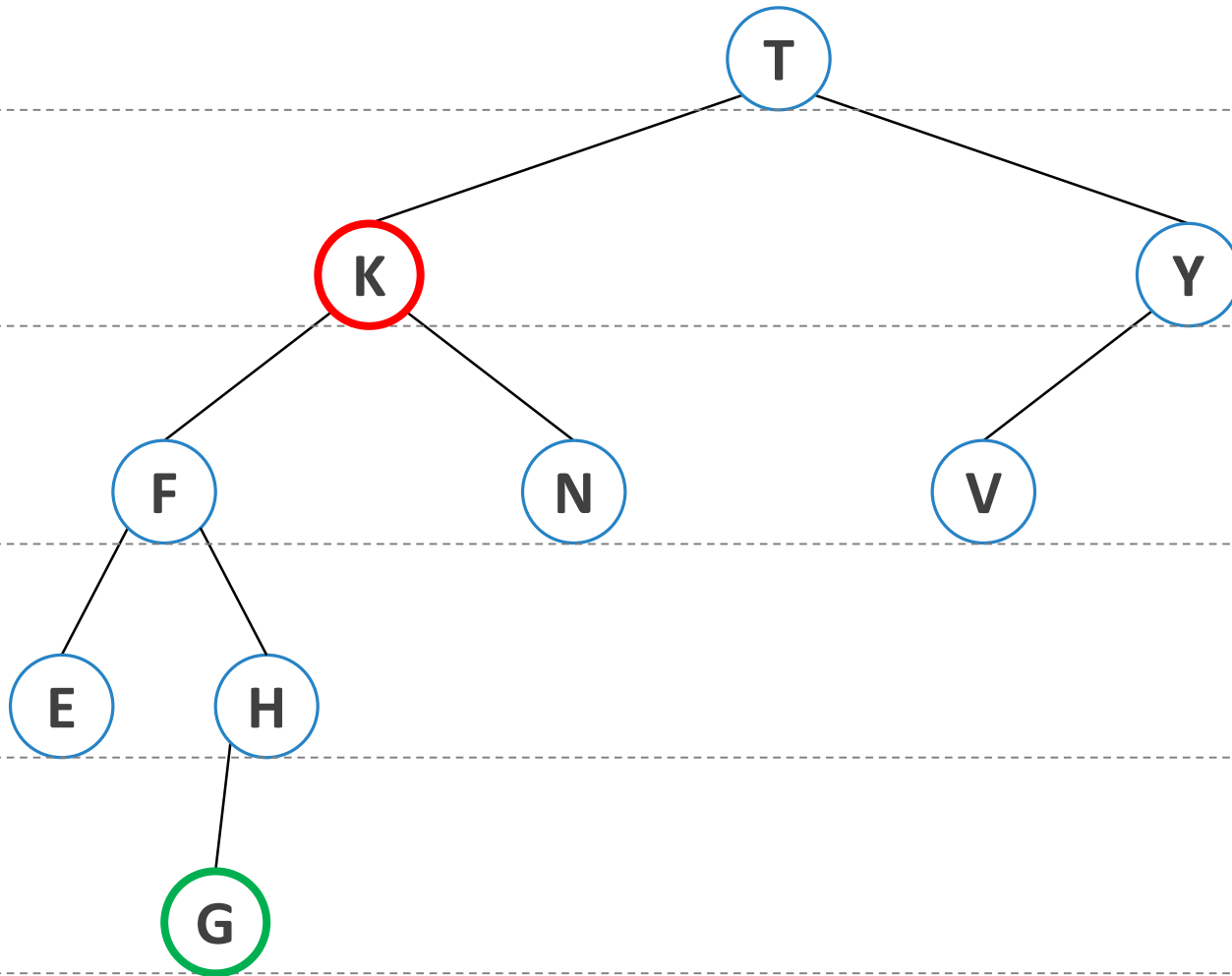
Entremos a T_2



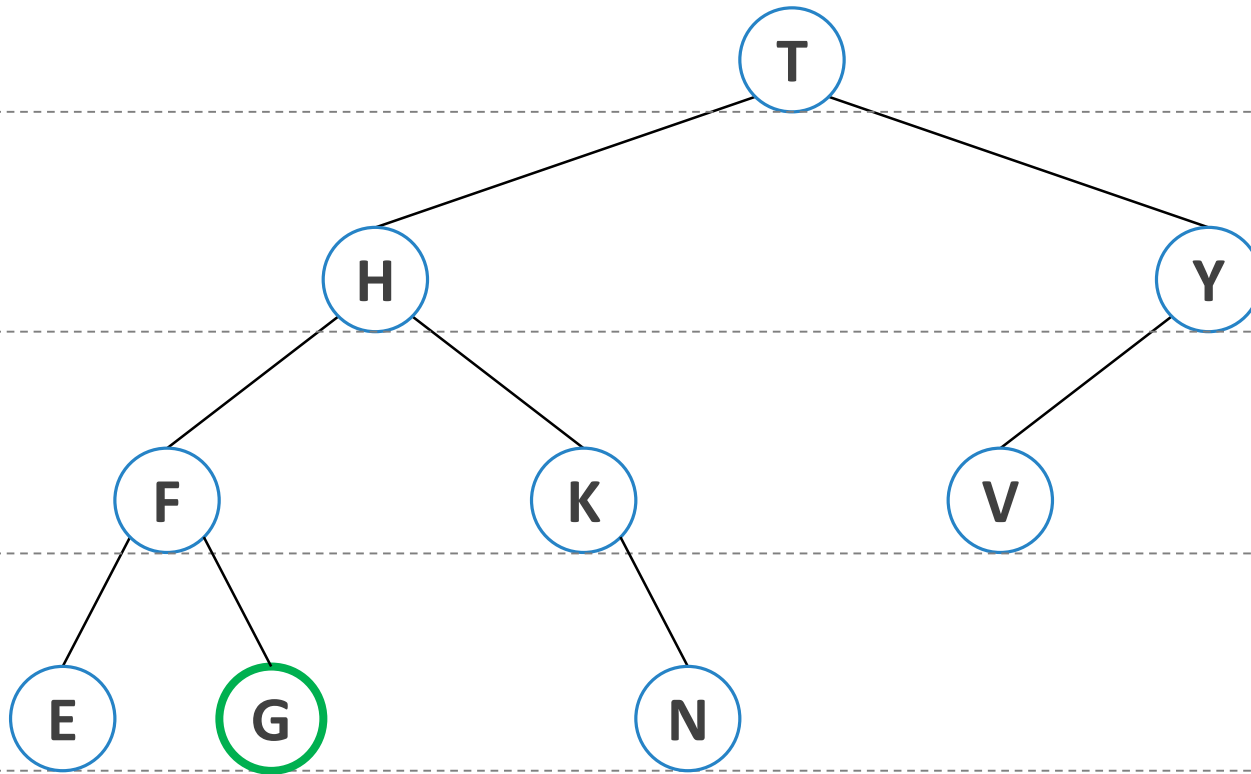
Doble rotación: primero a la izquierda en torno a Y-Z; luego a la derecha en torno a X-Z



Luego de insertar *G*



¡Doble rotación!



Propiedades de las rotaciones



¿Qué tan costoso es rebalancear el árbol?

¿Cuántas rotaciones es necesario hacer en el peor caso?

Altura de un árbol AVL



La complejidad sigue dependiendo de la altura del árbol

¿Pero cuál es la altura de un árbol AVL en el peor caso?