

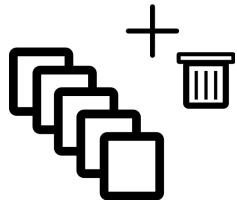
Árboles binarios

CI-0112 Programación 1

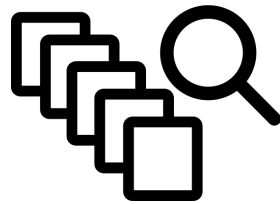
Sivana Hamer - sivana.hamer@ucr.ac.cr
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
Licencia: CC BY-NC-SA 4.0



¿Qué problemas tienen las estructuras de datos anteriores?



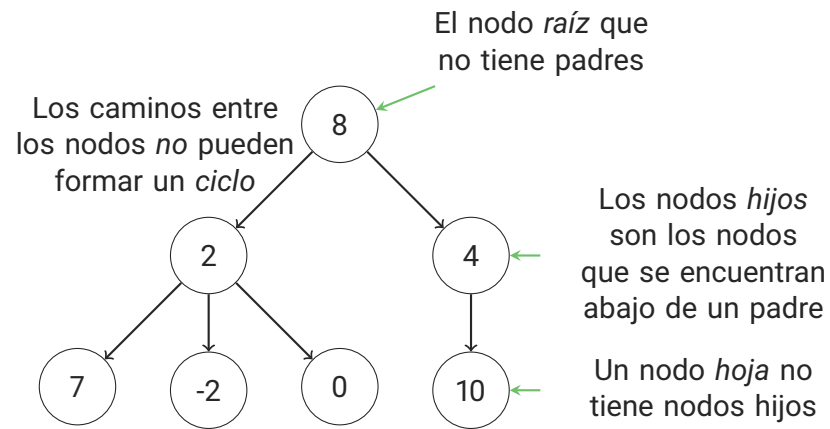
Insertar o borrar en arreglos requiere copiar los datos un nuevo espacio de memoria



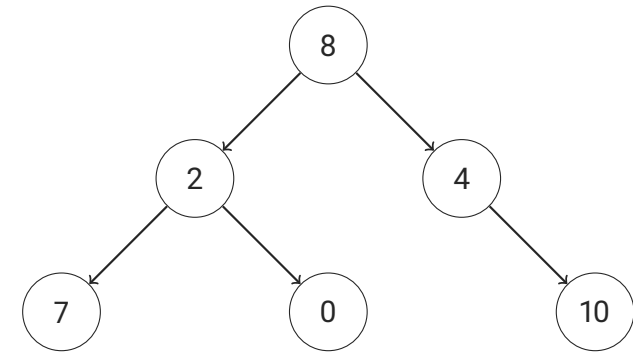
Acceder a elementos en listas requiere recorrer todos los elementos



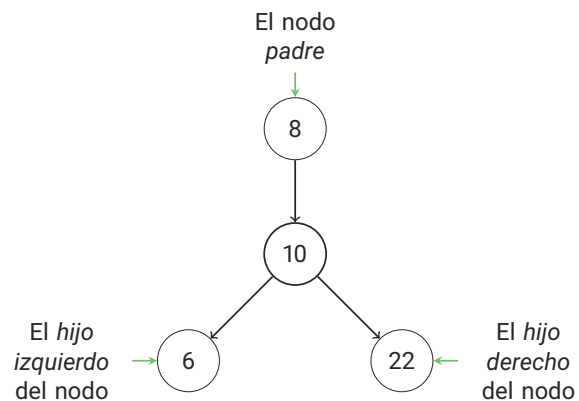
Un árbol es una estructura de datos que guarda nodos que tiene padres e hij@s



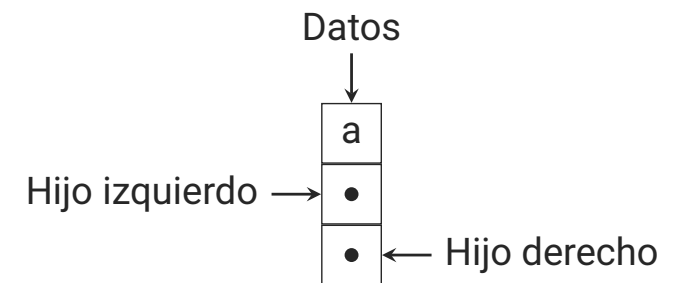
Un árbol binario tiene como máximo dos hij@s.



Un nodo de un árbol binario es el objeto que guarda el árbol, con referencias a los nodos hij@s

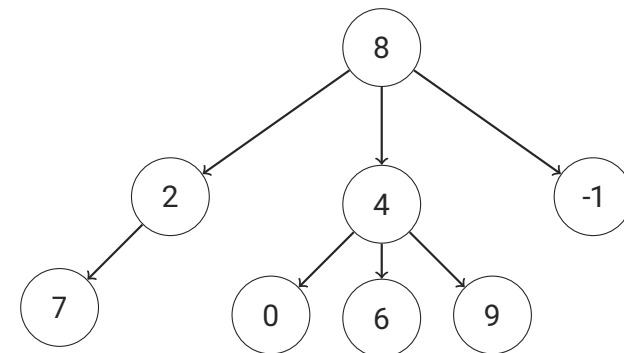


Cada nodo de un árbol guarda dos referencias, uno para el hijo izquierdo y otro para el derecho

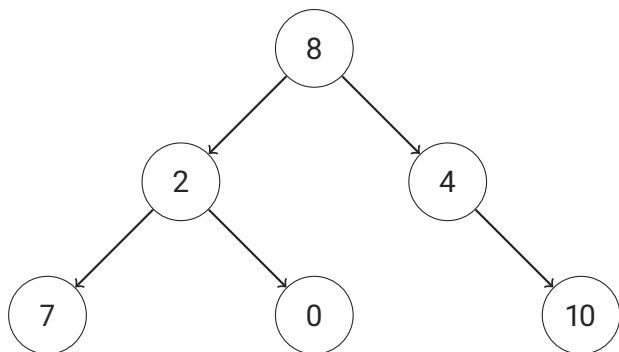


Existen distintos tipos de árboles...

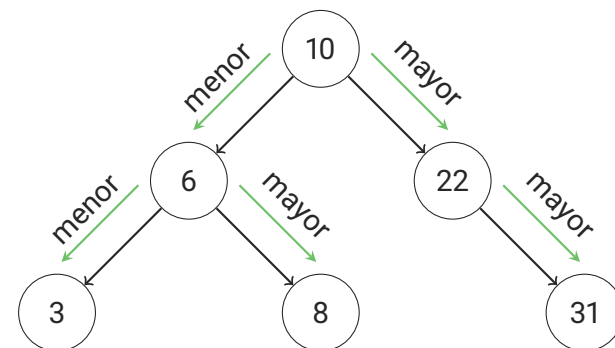
Un árbol n-ario tiene como máximo n hijos.



Un árbol balanceado tiene una cantidad similares de nodos en el lado derecho e izquierdo



Un Binary search (sort) tree (BST) agrega los elementos en orden, los descendientes izquierdos son menores que el nodo actual pero mayores a los descendientes derechos.

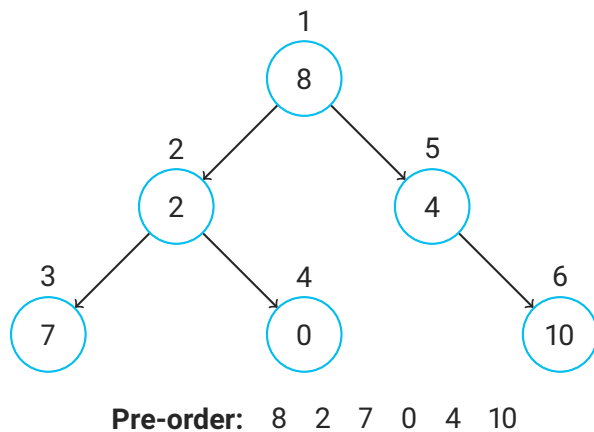


Dado la estructura del árbol, se puede recorrer de distintas maneras, dependiendo el orden en que se visitan los nodos

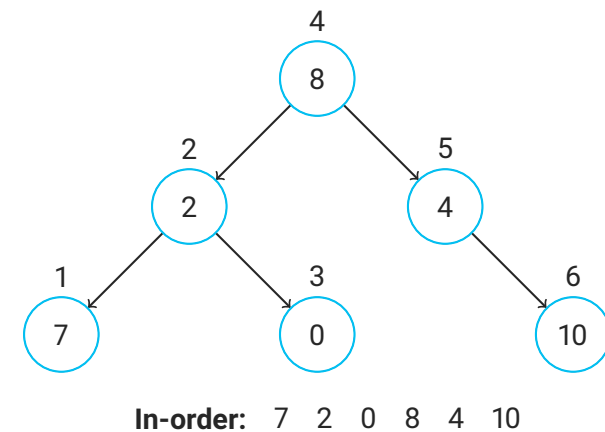
Veamos algoritmos para árboles...



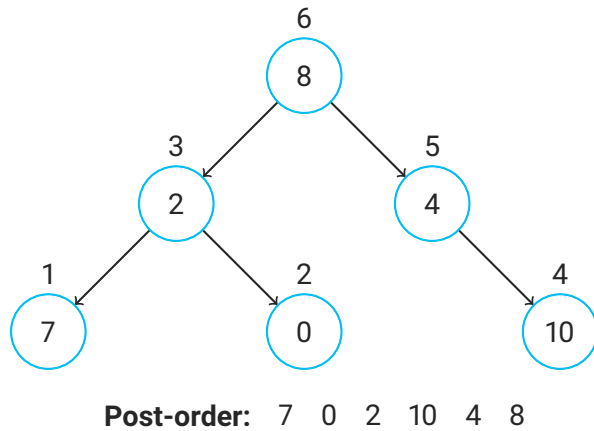
Pre-order busca en (1) Nodo actual (2) Hijo izquierdo (3) Hijo derecha



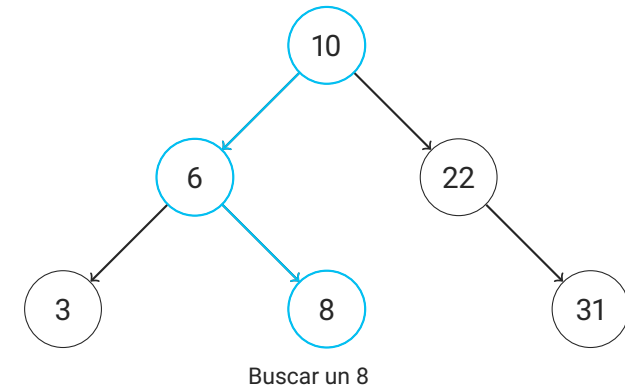
In-order busca en (1) Hijo izquierdo (2) Nodo actual (3) Hijo derecha



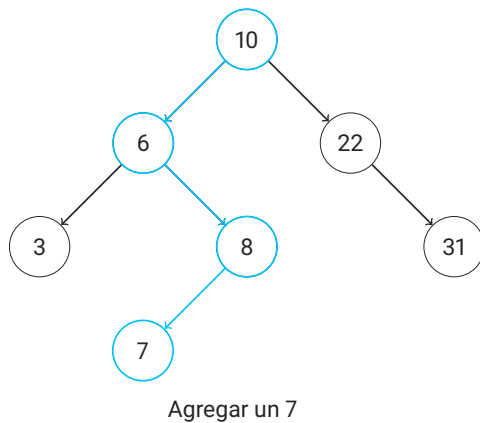
Post-order busca en (1) Hijo izquierdo (2) Hijo derecha (3) Nodo actual



Se puede *buscar* nodos en un BST rápido dado que tienen orden

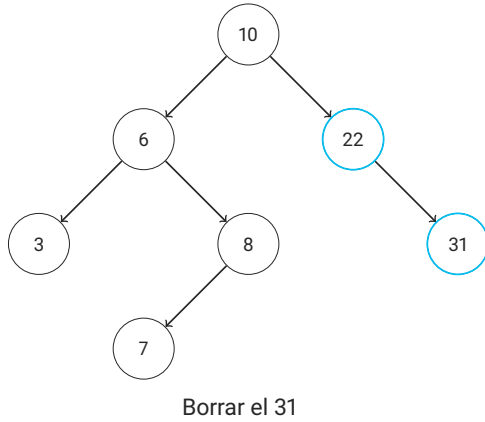


Para *insertar* en un BST debe seguir el orden, agregando el nodo donde se encontraría al buscar

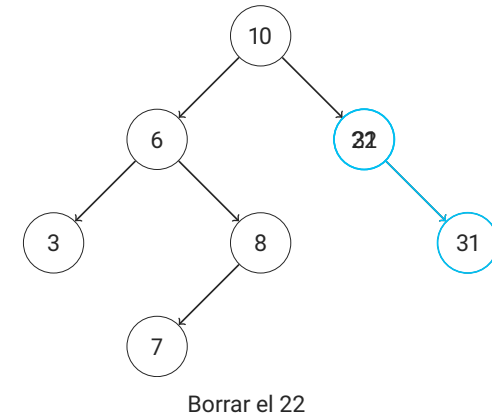


Para remover hay que pensar en diversos casos...

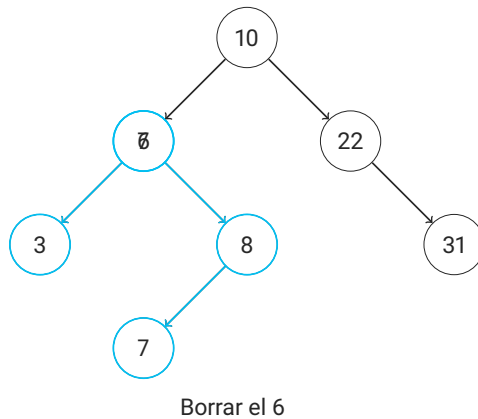
Para *remove* una *hoja* en un BST solo se quita la referencia del padre.



Para *remove* un *nodo con un hijo* en un BST el hijo toma la posición del padre



Para *remove* un *nodo con varios hijos* en un BST se reemplaza con el nodo más cercano (el más pequeño en el subárbol derecho)

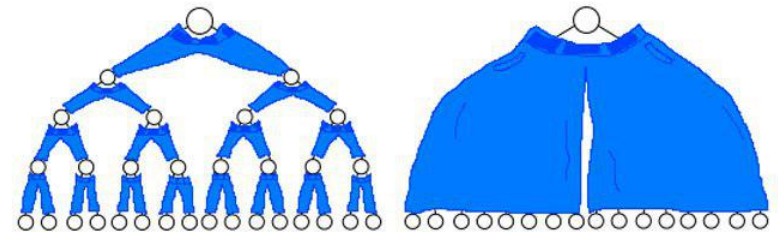


If a binary tree wore pants would he wear them

like this

or

like this?



Referencias I

- D. J. Eck, *Introduction to Programming Using Java*, eighth edition ed. Geneva (NY): Hobart and William Smith Colleges, Department of mathematics and computer science, 2020.
- G. L. McDowell, *Cracking the coding interview: 189 programming questions and solutions*. CareerCup, 2019.
- P. Morin, *Open Data Structures (in Java)*, 2013.
- mebeim. Can't really wrap my head around this one. [Image]. [Online]. Available: https://www.reddit.com/r/ProgrammerHumor/comments/6lvdth/cant_really_wrap_my_head_around_this_one/
- [Image]. [Online]. Available: https://static.wikia.nocookie.net/disney/images/2/22/The_Best_of_Both_Worlds.png/revision/latest?cb=20200726165336&path-prefix=es
- E. Demaine. 6. binary trees, part 1. [Video]. [Online]. Available: <https://www.youtube.com/watch?v=76dhtgZt38A>

Referencias II

- V. Costan. R5. recursion trees, binary search trees. [Video]. [Online]. Available: <https://www.youtube.com/watch?v=r5pXu1PAUkl>