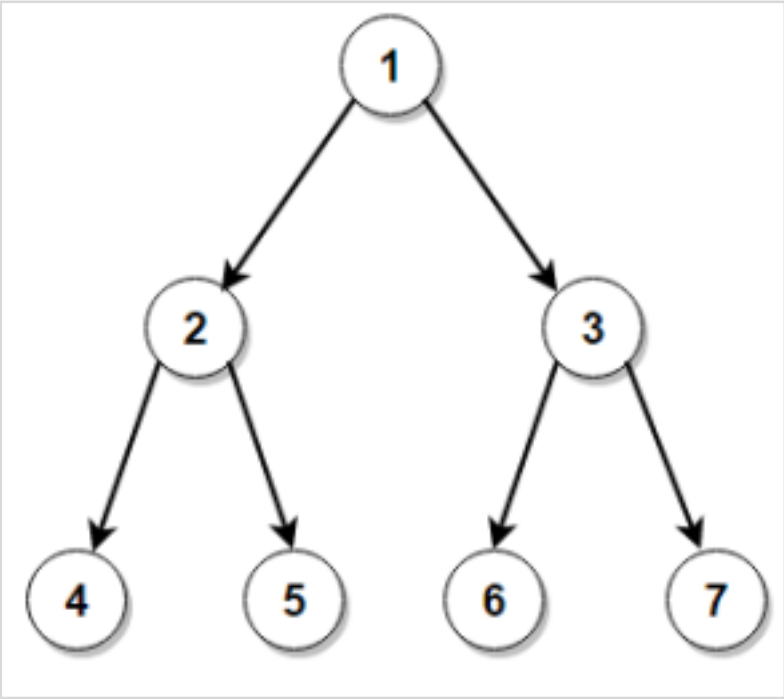
CS2100-Trees

#CS2100/individual

Implementar Arbol



Base Code

```
template <typename T>
class Node{
public:
    T key;
    Node<T> *left, *right;

    Node(T key){
        this->key = key;
        this->left = this->right = nullptr;
    }
};

template <typename T>
```

```
class Tree{
public:
  Node<T> * root;

  Tree(void):root(nullptr){
  }
};
```

Implementar las siguientes funcionalidades

- [ ] Check if two given binary trees are identical or not (1pts)
  - Identical Structure and Content
- [ ] Insert Node (2pts)
- [ ] Delete Node (2pts)
- [ ] Calculate height of a binary tree (1pts)
  - Number of nodes from root to leaf in the longest path
- [ ] Delete given Binary Tree (1pts)
  - Delete entire tree
- [ ] In-order Tree Traversal (1pts)
  - Left - Node - Right
- [ ] Pre-order Tree Traversal (1pts)
  - Node - Left - Right
- [ ] Post-order Tree Traversal (1pts)
  - Left - Right - Node
- [ ] Find next node in same level for given node in a binary tree (1pts)
- [ ] Check if given binary tree is complete binary tree or not (1pts)
  - Is call complete if every level, except the last, is completely filled and all nodes are as far left as posible
- [ ] Print all paths from root to leaf nodes in given binary tree (1pts)
- [ ] Find ancestors of given node in a Binary Tree (1pts)
- [ ] Check if given binary tree is height balanced or not (1pts)
  - Difference of length of height of right and left side are zero
- [ ] Determine if given Binary Tree is a BST or not (1pts)
- [ ] Convert a Binary Tree to BST by maintaining its original structure (2pts)
- [ ] Calculate height of a binary tree with leaf nodes forming a circular doubly linked list (2pts)

Tooling

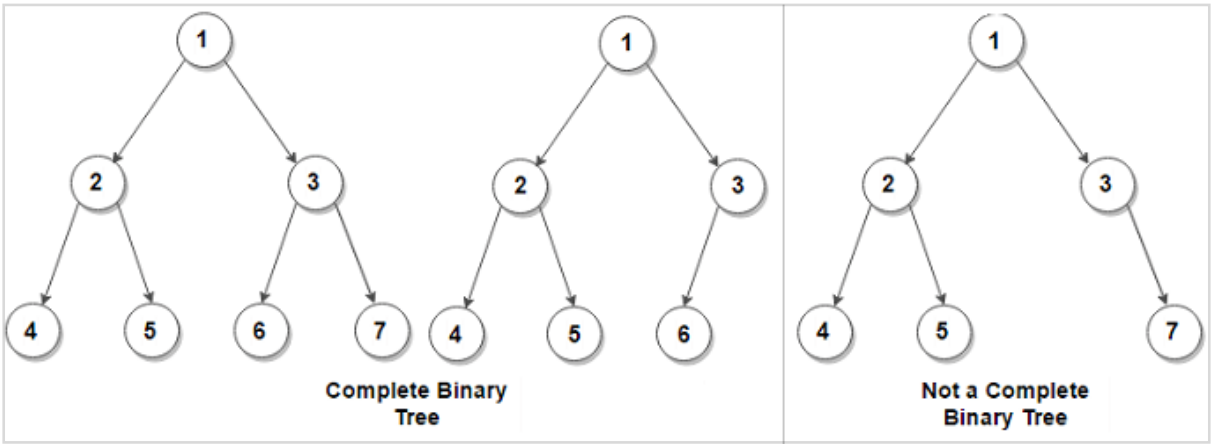
- Lenguaje C++
- Git + GitFlow + Karma Comments
- Si el código no compila se calificara sobre 11.
- Hacer un main para probar los ejercicios
- Evitar warnings
- Código limpio ➜ *Keep it short and simple*.

Help!

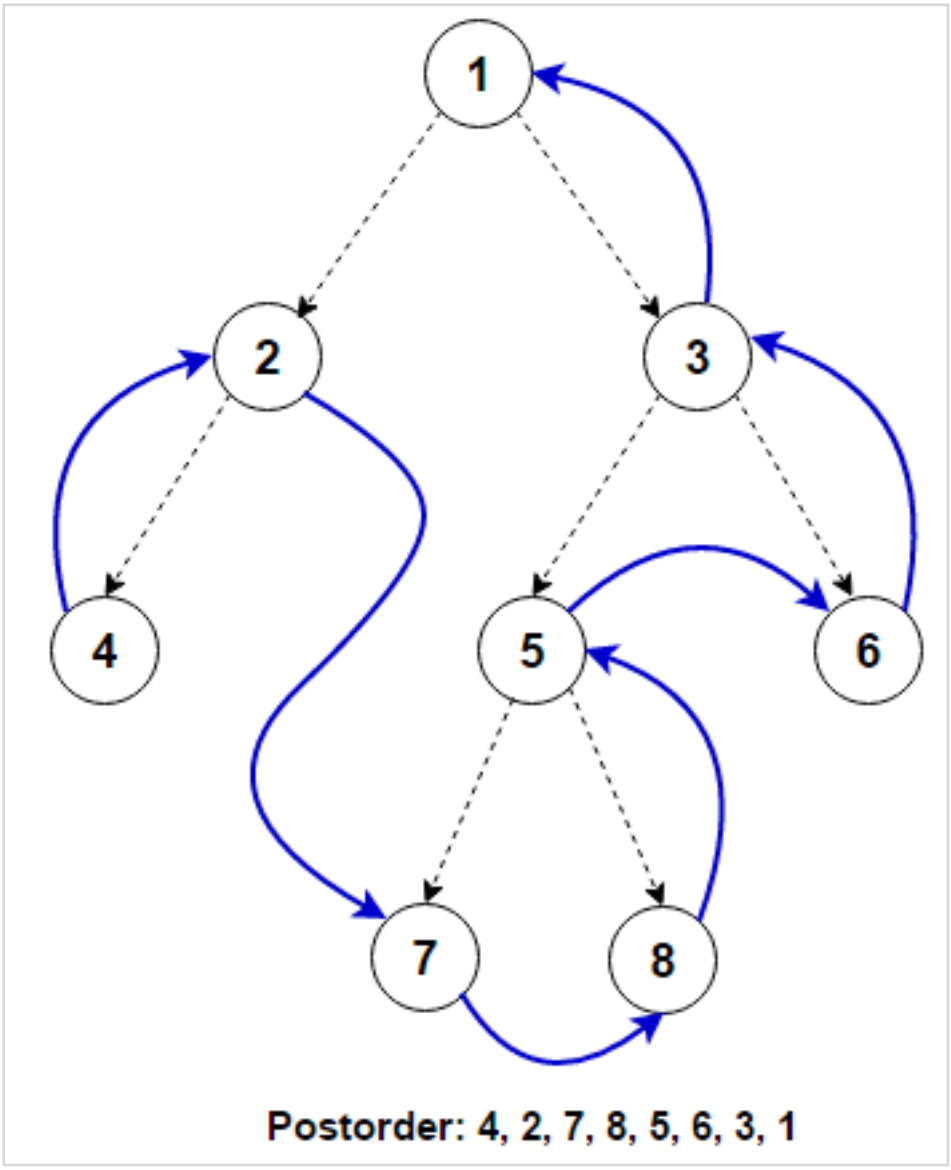
- Usemos el canal de Slack! (https://bit.ly/2LYbIbh)

Images

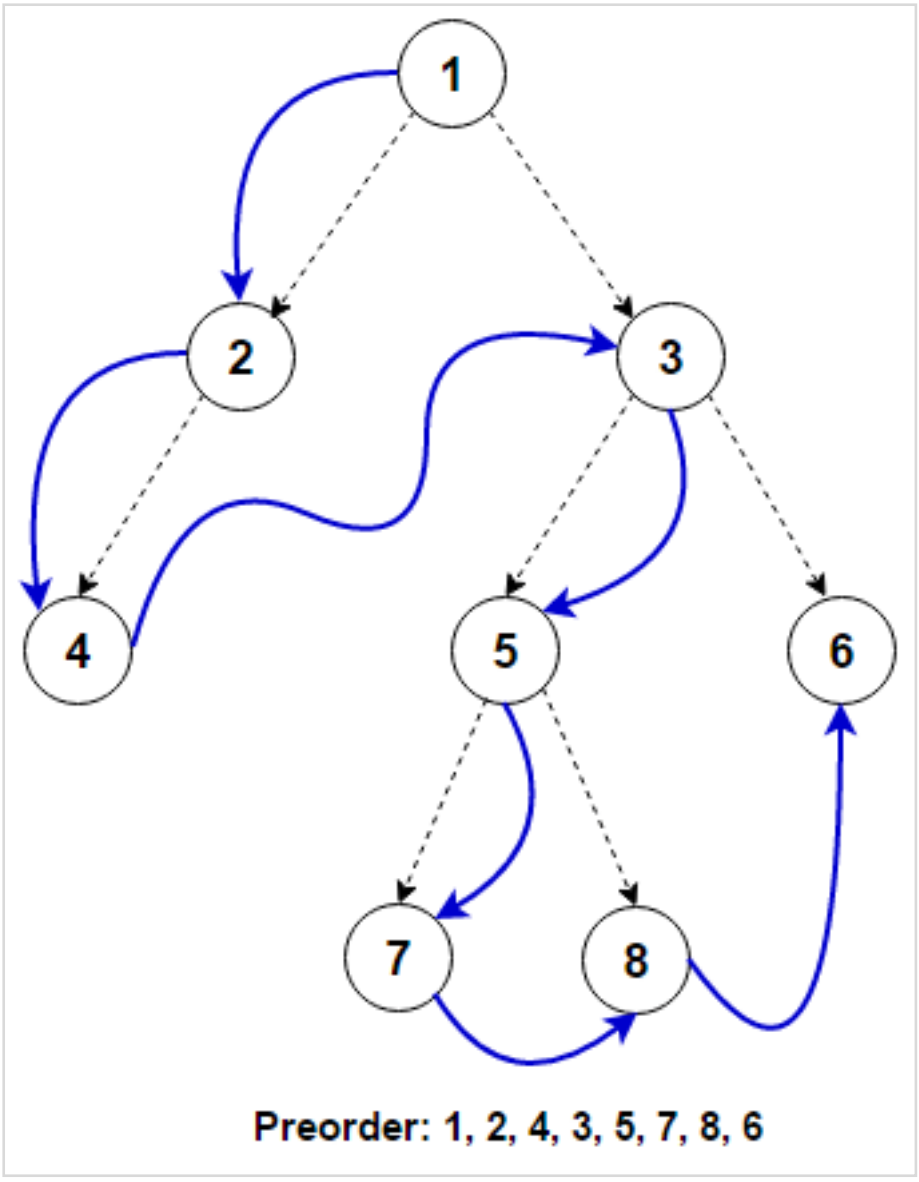
Complete Tree



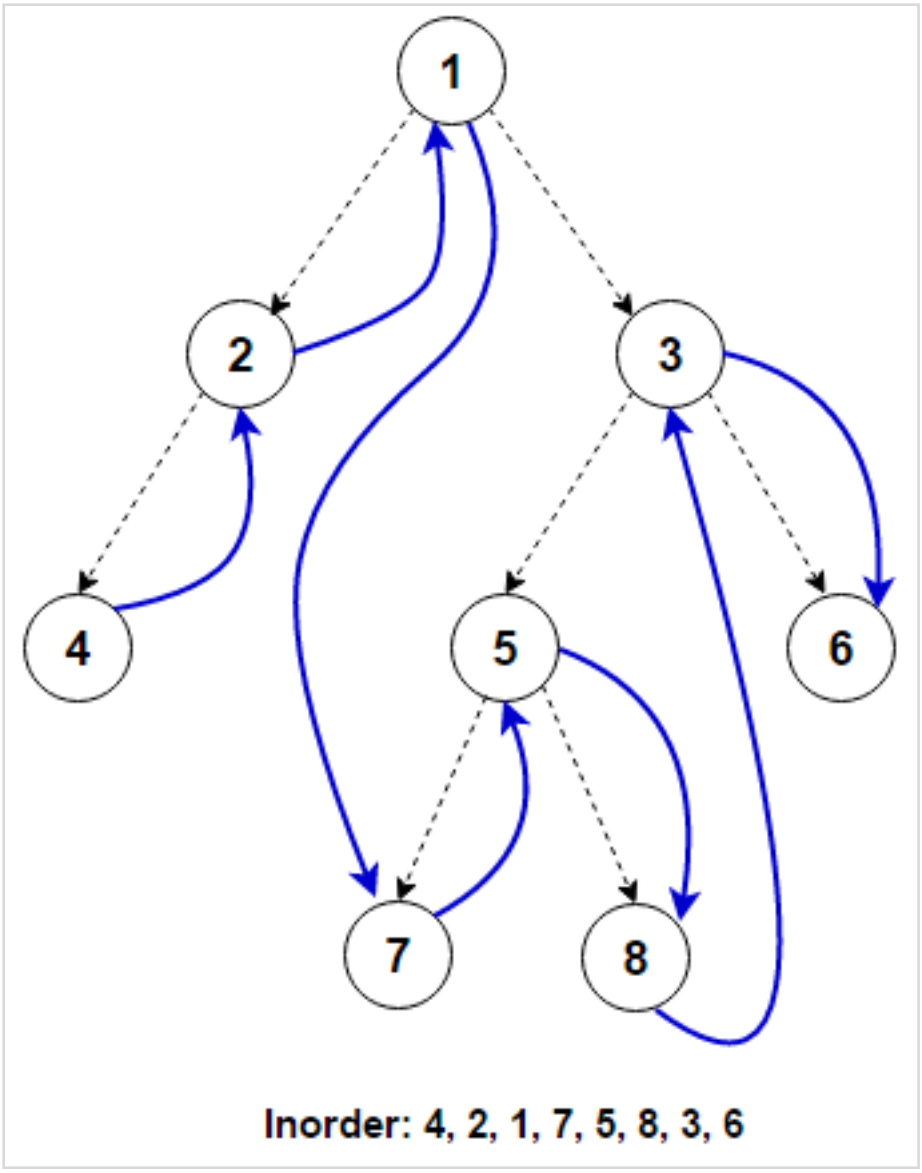
Post Order



Pre Order



In Order


1
2
3
4
5
6
7
8
Inorder: 4, 2, 1, 7, 5, 8, 3, 6