9.1

- Check if bin. tree is height-balanced

Input: root of bin. tree
Output: bool

I probably have to check the children of each node.

Use tree traversal?

→ Use recursion

§ #9

- perfect binary tree of height h contains exactly $2^{h+1}-1$ nodes. of which 2^h are leave

- Complete binary tree on n nodes has height $\lfloor \log n \rfloor$.

- Inorder, preorder, post order traversals.

Let T be a Binary Tree of n nodes w/ height h.
Implemented recursively, these traversals have $O(n)$ time and $O(h)$ space

Is tree balanced

9.1 7/13

Brute force

- Starting from the leaves, compute the height for each node proceeding upwards.
- Check if height of left and right children are greater than one.
- Store heights in a hash table
- Recursive dfs?
- O(n) time, space where n is number of nodes

Better approach

- We donot need to store the heights, just need to know if each node is height balanced. if yes, know the height and use it to check the height for the ~~next~~ parent node.

- To check if balanced, we need to know if node in left and right is balanced and want to know the height. (height ≤ 1?) for both left and right.
- We can know ~~that~~ height when we are making the recursive calls (check_balanced) which is essentially traversing the tree.
  - ↳ The height is the max height of (left and right node) + 1.