1a)A stack is an abstract data structure that follows the Last In, First Out principle - the most recently "pushed" element is the first one to be removed or "popped". This means that the sequential ordering of the elements is preserved

A stack is useful for solving this problem because it allows the tracking of opening brackets as they are pushed on and verifies that any following closing brackets match the most recent opening parenthesis at the top of the stack.

This process guarantees that nested and sequential parentheses are paired correctly.

Once the string has been processed:

- If the stack is empty, all parentheses were matched correctly, meaning the string is valid.
- If the stack is not empty, there are unmatched opening parentheses, meaning the string is invalid.

Each character is processed exactly once and the push and pop operations are fast.

Proof : Prove that any (non-emty) perfectly balanced binary tree of height h has exacty $2^{h+1}-1$ nodes.

Base case : When $h=0$ (i.e when the tree only has a root) the number of nodes is 1

When $h=0$, $n = 2^{h+1}-1 = 2^{0+1}-1 = 2^{1}-1 = 2-1 = 1$

where n = number of nodes.

Inductive hypothesis : Assume the rule is true for some positive integers k (where k is the height)

$$n_k = 2^{k+1}-1 \quad \text{where } n = \text{number of nodes.}$$

Inductive Step : Prove the rule holds true for every "next step" of k - i.e prove rule holds true for heights of k+1.

→ At height k the tree has 2^k nodes at the bottom

→ Therefore, at height k+1 the tree will have 2^{k+1} nodes at the bottom as thes doubles the number of bottom nodes

→ The total number of nodes at k+1 height will be:

$$n_{k+1} = n_k + 2^{k+1}$$

Substitute equation for n_k:

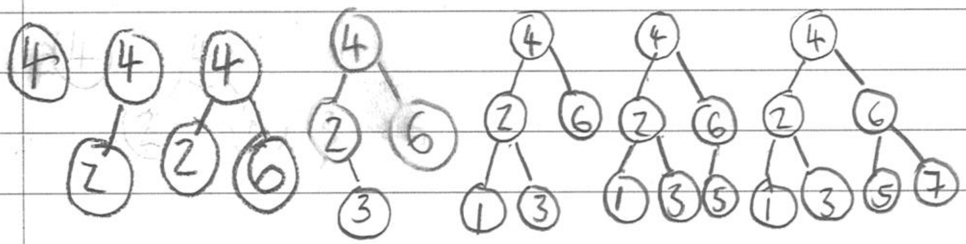
$$n_{k+1} = 2^{k+1} - 1 + 2^{k+1}$$
$$= 4^{k+1} - 1$$
$$2^{1} \cdot 2^{k+1} - 1$$
$$= 2^{(k+1)+1} - 1$$

which is equal to substituting k+1 into the rule:

$$n_{k+1} = 2^{(k+1)+1} - 1$$

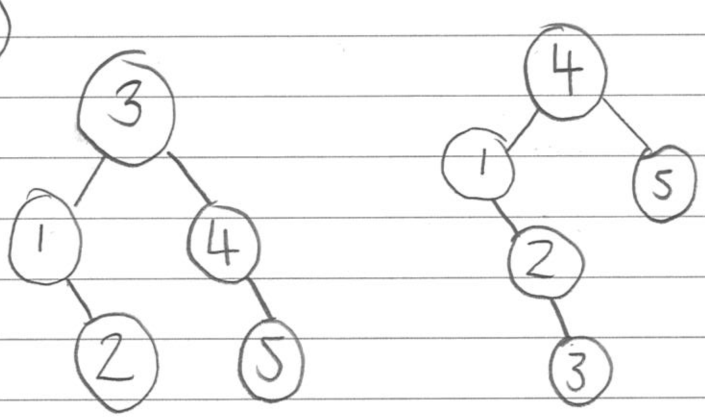
where $k+1 = h$,

therefore, by induction, the rule holds true for all posivitive integers b)1) (4, 2, 6, 3, 1, 5, 7)



b)2) (4, 2, 6, 7, 5, 1, 3)

c)


3
1
4
2
5
4
1
5
2
3