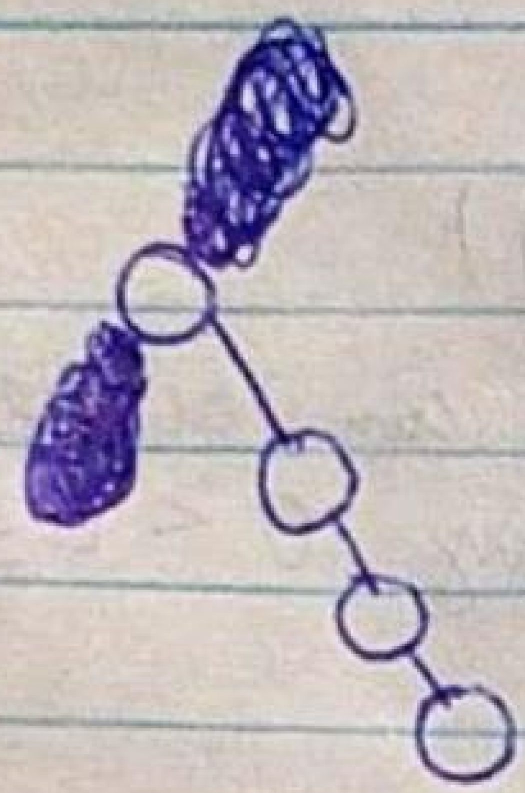
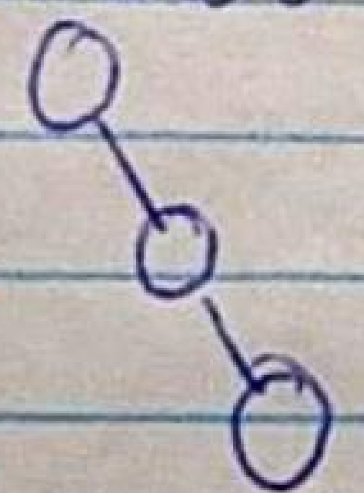
2\. a 1. Counter - example to disprove :
Consider the BST T :



For every node $u \in T$, we can choose $c > 0$ st. $H(left(u)) \leq H(right(u)) + c$.
However the height of the tree is $O(n)$ not $O(\log n)$.

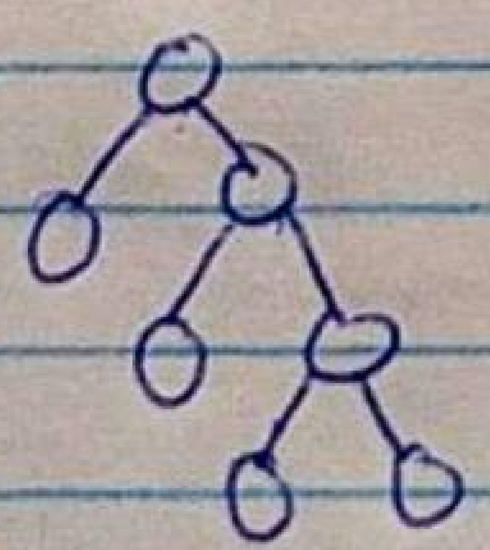
2\. Counter - example:
Consider BST T of size $n = 3$:



For every node $u \in T$, we can choose $c > n$ st
$H(left(u)) - c \leq H(right(u)) \leq H(left(u)) + c$.

However the height of the tree is $O(n)$ not $O(\log n)$.

3\. Counter - example: Consider BST T of size $n = 7$.



Every internal node $u \in T$ has exactly 2 children. However height of BST is $O\left(\frac{n-1}{2}\right) = O(n)$ and not $O(\log n)$.

b For a BST to have height $O(\log n)$, it must be complete / balanced.

If we choose $c = 1$, $[0 < c \leq 1]$
$\forall u \in T$,

$N(left(u)) \geq \overset{1}{c} \cdot N(right(u)) - 1$

$\Rightarrow N(left(u)) \geq N(right(u)) - 1$ (1)

Similarly $N(right(u)) \geq N(left(u)) - 1$ (2)

From (1) & (2),

$N(left(u)) = N(right(u))$

$\Rightarrow$ The tree is complete. We can similarly prove this for any $0 < c \leq 1$.

or $h \in O(\log n)$.