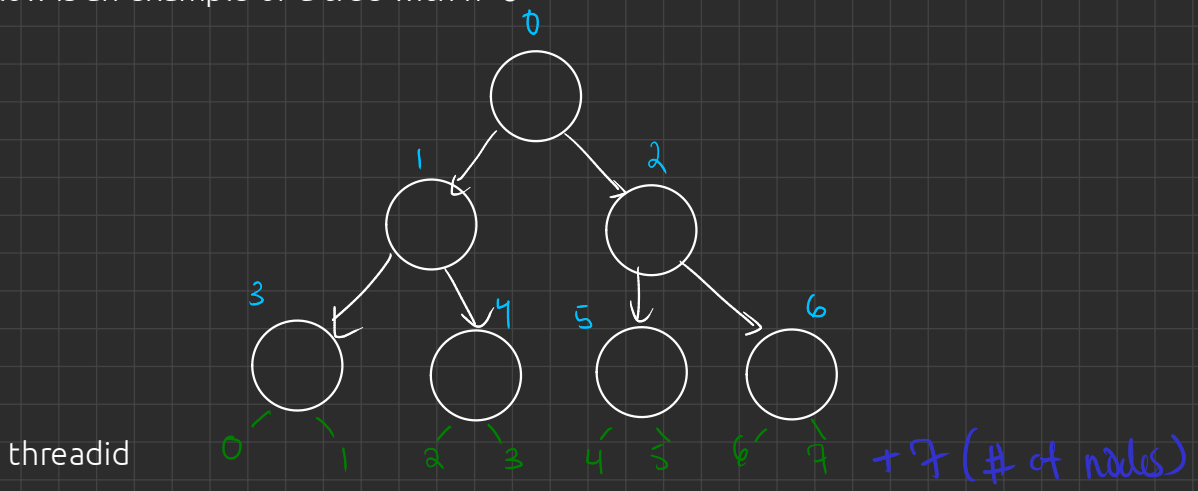
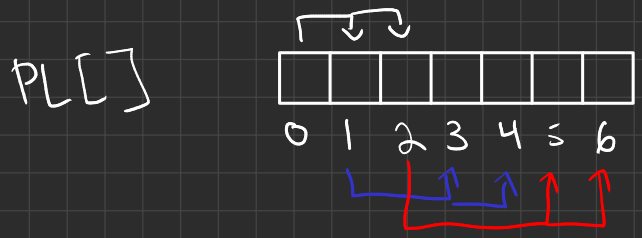
Tournament Tree Algorithm using Peterson Locks

For each lock, there is 2 processes at most competing for the lock
Each lock there is basically a normal Peterson Lock.
If the processes wins the lock, it will obtain permission to go to the next lock
There will always be n-1 or n amount of lock nodes depending on parity of the number of threads.
if thread number == even: n -1
if thread number == odd: n
This will be an implicit type of binary tree.
Therefore, an array will be used and indexes calculated to determine the left child and right child, as well as the parent child.
Below is an example of a tree with n=8

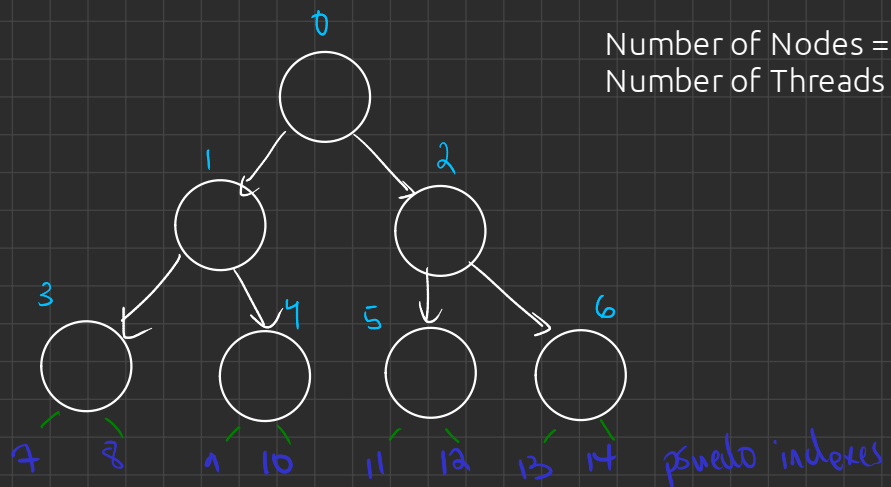




Therefore, formula for nodes is as follows:
Left Child: 2n + 1
Right Child: 2n + 2
Parent: $\lfloor \frac{n-1}{2} \rfloor$

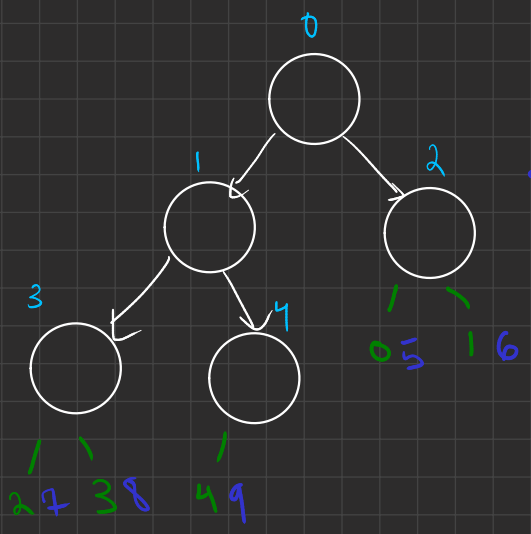
Where n is the index of the current lock node.
When using thread ids that start at 0,
to find which lock the thread is associated with,
simply add n to the thread id and perform
the parent formula.

Example of assigning pseudo indexes to processes



Number of Nodes = 7
Number of Threads = 8

Example of another type of tree



• - thread ids
• - pseudo node indexes

Number of Nodes = 5
Number of Threads = 5