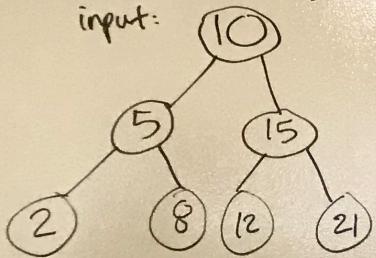


Problem Domain

Write a breadth first traversal method which takes a Binary Tree as its unique input, traverse the tree using breadth first, and print every visited nodes value.

input:



output: 10, 5, 15, 2, 8, 12, 21

Big O()
time $O(n)$
space $O(n)$

Algorithm

Given a Binary Tree as input,
Create a queue that has all
the standard methods.
Enqueue the root node. ← create a current node
create an output string
While the queue has a valid node
if binary tree empty) Set current to front node value
of queue (dequeue)
if current node has a left child
enqueue current node left child
if current node has a right child
enqueue current node right child
assign queue.dequeue to temp node
add temp node value to output string
return

Edge case

- Binary tree is None
returns empty string

Code

```
from stack_and_queue import Queue  
def breadth_first_traversal(binary_tree):  
    queue = Queue()  
    if binary_tree.root_node is None:  
        return output  
    output = ""  
    queue.enqueue(binary_tree.root_node)  
    temp = Node()  
    current = Node()  
    while queue.peek():  
        current = queue.dequeue()  
        if current.left_child:  
            queue.enqueue(current.left_child)  
        if current.right_child:  
            queue.enqueue(current.right_child)  
        temp = queue.dequeue()  
        output += temp.value  
    return output
```