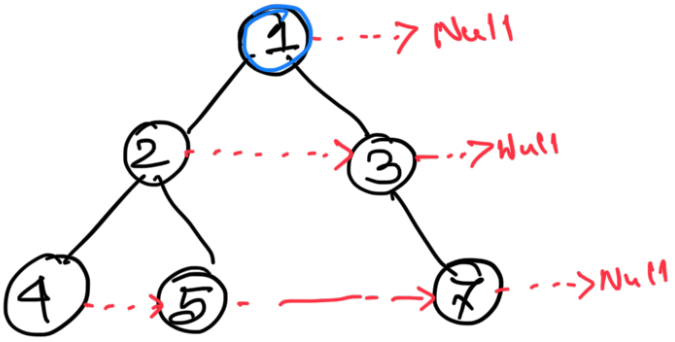
# Populate Next Right Pointer In Each Node Part 2

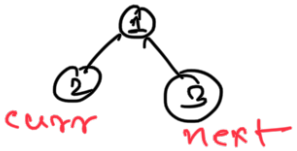



* First Approach using BFS where

T.C. = O(n)
S.C. = O(n)

## [BFS solution]

**Idea:** Basically we are going to use Two Pointer approach here. We will run BFS, and apply two pointer, One is the current root and another is next which is the sibling node of that level.




we need to link between 2 ---> 3

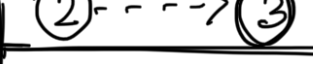
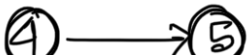
curr.next = next

Initially queue = [1]

```
queue.add(root);
while (!queue.isEmpty()){
    Node curr = null; Node next = null;
    int qLen = queue.size();
    for(int i=0; i< qLen; i++) {
        if (curr == null){
            curr = queue.poll();
            append left & right child in the queue
        }
        else if(next == null) {
            next = queue.poll();
            curr.next = next;
            append left & right child in the queue
        }
        else{
            curr = next
            poll();
```

next = queue.poll();
curr.next = next;
append left & right child in the queue

} } }

| while | range | queue | curr | next | |
|---|---|---|---|---|---|
| 1 | | [1] | Null | null | |
| 1 | 0<1 | [~~1~~, 2, 3] | 1 | null | |
| 2 | | [2, 3] | null | null | |
| 2 | 0<2 | [~~2~~, 3, 4, 5] | 2 | null | |
| 2 | 1<2 | [~~3~~, 4, 5, 7] | 2 | 3 | new we have our next. curr.next = next  |
| 3 | | [4, 5, 7] | null | null | |
| 3 | 0<3 | [~~4~~, 5, 7] | 4 | null | |
| 3 | 1<3 | [~~5~~, 7] | 4 | 5 |  |
| 3 | 2<3 | [~~7~~] | 5 | 7 |  |

curr = next

next = queue.poll();

curr.next = next

This is the corner case, if our curr and next both is non-null, the we will swap them. like

curr = next;

Taking the new next from the queue;

next = queue.poll();

Stablish the link between the as before;

curr.next = next;