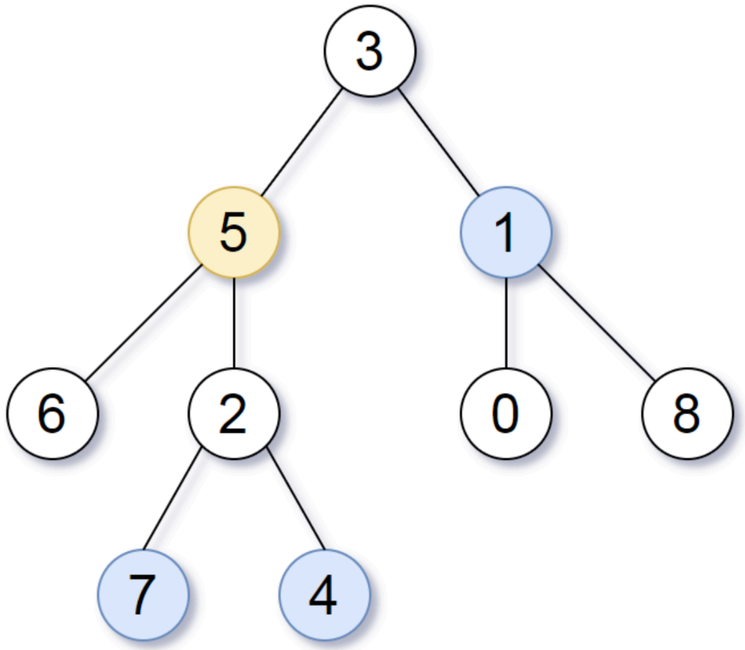
863. All Nodes Distance K in Binary Tree

Given the `root` of a binary tree, the value of a target node `target`, and an integer `k`, return *an array of the values of all nodes that have a distance* `k` *from the target node.*

You can return the answer in **any order**.

Example 1:



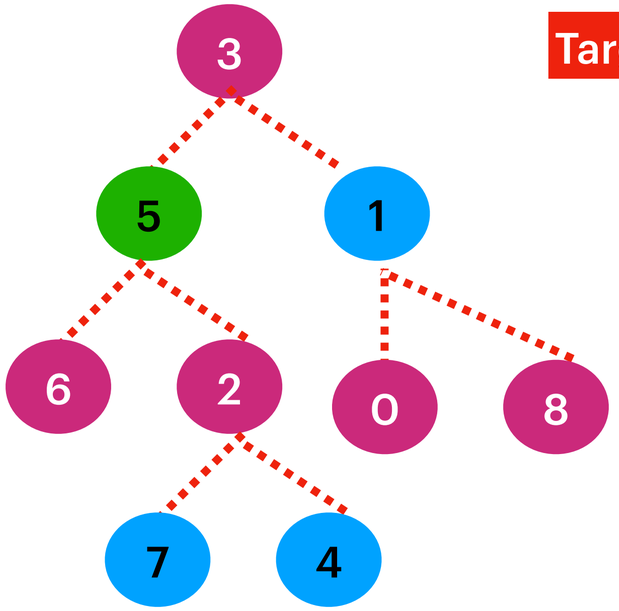

```
Input: root = [3,5,1,6,2,0,8,null,null,7,4], target = 5, k = 2
Output: [7,4,1]
Explanation: The nodes that are a distance 2 from the target
node (with value 5) have values 7, 4, and 1.
```

Example 2:

```
Input: root = [1], target = 1, k = 3
Output: []
```

Constraints:

- The number of nodes in the tree is in the range `[1, 500]`.
- `0 <= Node.val <= 500`
- All the values `Node.val` are **unique**.
- `target` is the value of one of the nodes in the tree.
- `0 <= k <= 1000`



Target = 5 , K=2

863. All Nodes Distance K in Binary Tree

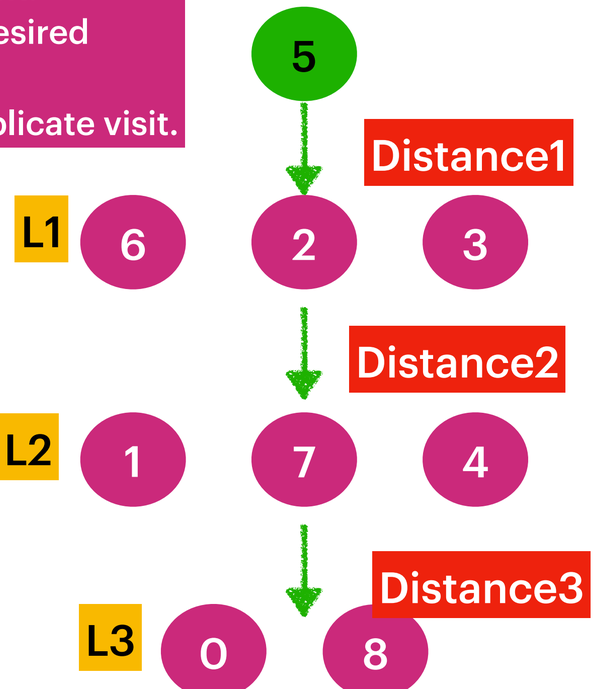
Distance 1 For Target 5 => 6 2 3

Distance 2 For Target 5 => 1 7 4

Distance 3 For Target 5 => 0 8

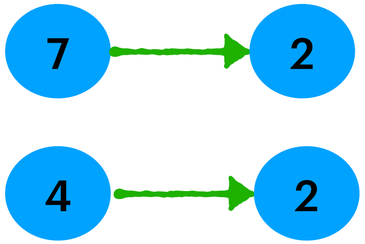
Distance >= 4 For Target 5 => []

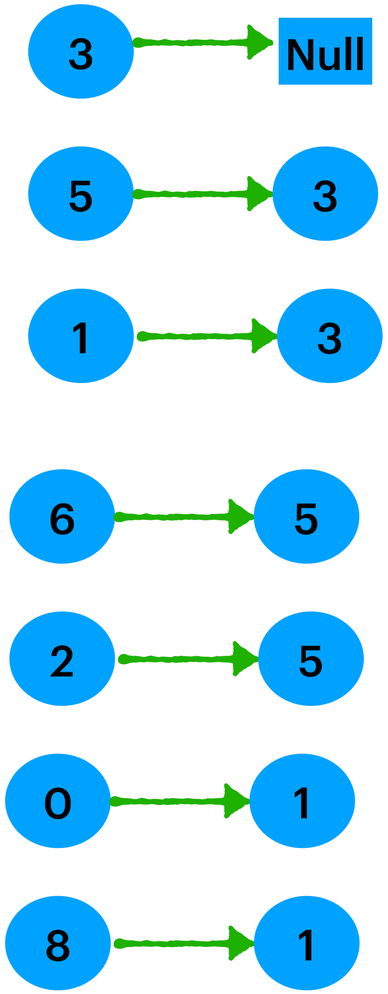
Algorithm :
Map each node to its Parent.
Use BFS from target to get desired Distance nodes.
Make use of visitedSet to avoid duplicate visit.



Time Complexity : O(n)
Space Complexity : O(n)

Map EachNode with its parent :





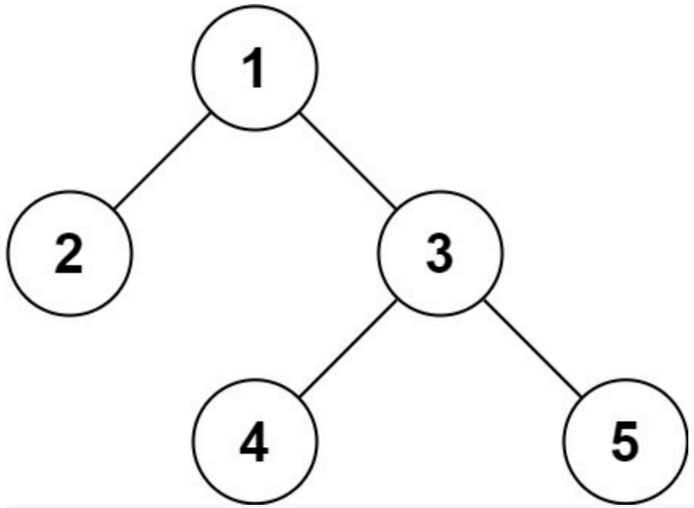
297. Serialize and Deserialize Binary Tree

Serialization is the process of converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment.

Design an algorithm to serialize and deserialize a binary tree. There is no restriction on how your serialization/deserialization algorithm should work. You just need to ensure that a binary tree can be serialized to a string and this string can be deserialized to the original tree structure.

Clarification: The input/output format is the same as how LeetCode serializes a binary tree. You do not necessarily need to follow this format, so please be creative and come up with different approaches yourself.

Example 1:




```
Input: root = [1,2,3,null,null,4,5]
Output: [1,2,3,null,null,4,5]
```

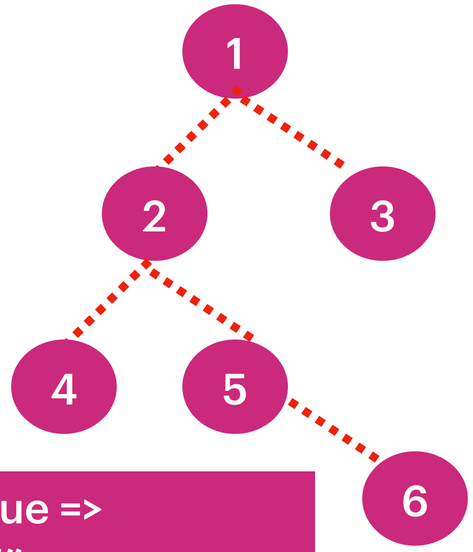
Example 2:

```
Input: root = []
Output: []
```

Constraints:

- The number of nodes in the tree is in the range `[0, 10^4]`.
- `-1000 <= Node.val <= 1000`

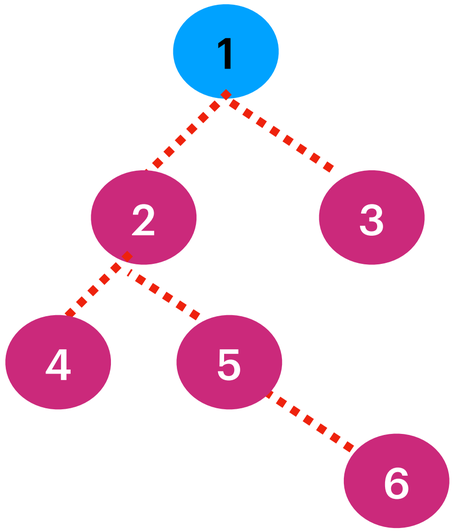
297. Serialize and Deserialize Binary Tree



Algorithm :
Serialize BFS Travel
[Handle null leff/right with helper value]
Deserialize BFS Travel

Serialisation : returns String Value =>
“1, 2, 3 , 4, 5, E, E,E,E,E,6,E,E”
DeSerialisation get the tree from above Serialised String.

“1, 2, 3 , 4, 5, E, E,E,E,E,6,E,E”



Time Complexity : O(n)
Space Complexity : O(n)