

Leetcode-104 - Maximum depth of Binary Tree.

① 이해	<p>* 자료구조: 이진트리,</p> <p>· 문제에서 주어진 이진 트리의 최대 깊이만을 구하면 되는 문제.</p>
② 계획	<p>· 이진 트리 역시 그래프의 일종이기 때문에, BFS/DFS로 풀이 가능.</p> <p>· 알고리즘: BFS → 큐 사용.</p>
③ 풀이	<p>1. BFS를 하기 위한 큐 생성</p> <p>2. 큐에 root node를 삽입</p> <p>3. 큐에 있는 노드를 pop하여 자식 노드가 있으면 자식 노드를 큐에 append { 없으면 pass 같은 레벨의</p> <p>3번 반복시 항상 제일 처음 만족된 시점에 큐의 원소 개수 만큼만 시행 (노드 개수 만큼 시행)</p> <p>4. 3번을 큐에 원소가 존재하지 않을 때까지 반복 (3번 시행에서 어느 노드도 자식 노드가 존재 X)</p>
④ 비교	<p>예제 코드) for _ in len(queue): ~</p> <p>↳ 만족문이 시행되어도 제일 처음 값을 유지.</p>
⑤ 최종 코드	<pre> class Solution: def maxDepth(self, root: Optional[TreeNode]) -> int: if root is None: return 0 ①, ② queue = collections.deque([root]) depth = 0 while queue: depth += 1 for _ in range(len(queue)): pop = queue.popleft() if pop.left: queue.append(pop.left) if pop.right: queue.append(pop.right) return depth </pre> <p>ex) </p>