

LeetCode Bootcamp Week 7 Submission
Submitted By: Raunak Choudhary (NetID: rc5553)

Question1: Binary Tree Right Side View

Code:

```
class Solution(object):
    def rightSideView(self, root):
        result = []

        def dfs(node, level):
            if not node:
                return

            if level == len(result):
                result.append(node.val)

            dfs(node.right, level + 1)
            dfs(node.left, level + 1)

        dfs(root, 0)
        return result
```

Question2: Rotting Oranges

Code:

```
class Solution(object):
    def orangesRotting(self, grid):
        rows, cols = len(grid), len(grid[0])
        fresh = 0
        queue = []

        for r in range(rows):
            for c in range(cols):
                if grid[r][c] == 2:
                    queue.append((r, c))
                elif grid[r][c] == 1:
                    fresh += 1

        if fresh == 0:
            return 0

        directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
        minutes = 0

        while queue and fresh > 0:
            minutes += 1
            size = len(queue)

            for _ in range(size):
                r, c = queue.pop(0)

                for dr, dc in directions:
                    nr, nc = r + dr, c + dc

                    if 0 <= nr < rows and 0 <= nc < cols and grid[nr][nc] == 1:
                        grid[nr][nc] = 2
                        fresh -= 1
                        queue.append((nr, nc))

        return minutes if fresh == 0 else -1
```

Question 3: Course Schedule II

Code:

```
class Solution(object):
    def findOrder(self, numCourses, prerequisites):
        graph = defaultdict(list)
        in_degree = [0] * numCourses

        for course, prereq in prerequisites:
            graph[prereq].append(course)
            in_degree[course] += 1

        queue = deque()
        for course in range(numCourses):
            if in_degree[course] == 0:
                queue.append(course)

        result = []
        while queue:
            course = queue.popleft()
            result.append(course)

            for next_course in graph[course]:
                in_degree[next_course] -= 1
                if in_degree[next_course] == 0:
                    queue.append(next_course)

        return result if len(result) == numCourses else []
```