EXPERIMENT - 10

Student Name: Yash Dixit **UID:** 22BCS10730
Branch: BE-CSE **Section/Group:** IOT-605‘A’
Semester: 6th **Date of Performance:** 02/04/25
Subject Name: Advance Programming **Subject Code:** 22CSP-367

✨ Miscellaneous Problems ✨

Day 21 (Miscellaneous - Easy)

1. Find Celebrity (Amazon)

Link

```
python
CopyEdit
def findCelebrity(n):
    def knows(a, b):
        # returns True if person a knows person b
        pass

    candidate = 0
    for i in range(1, n):
        if knows(candidate, i):
            candidate = i

    for i in range(n):
        if i != candidate and (knows(candidate, i) or not knows(i, candidate)):
            return -1
    return candidate
```

2. Pascal's Triangle (Apple)

Link

```
python
CopyEdit
def generate(numRows):
    triangle = []
    for i in range(numRows):
        row = [1] * (i + 1)
        for j in range(1, i):
            row[j] = triangle[i-1][j-1] + triangle[i-1][j]
        triangle.append(row)
    return triangle
```

3. Hamming Distance (LinkedIn)

Link

```
python
CopyEdit
def hammingDistance(x, y):
    return bin(x ^ y).count('1')
```

Day 22 (Miscellaneous - Medium)

4. Task Scheduler (Netflix)

Link

```
python
CopyEdit
from collections import Counter

def leastInterval(tasks, n):
    task_count = Counter(tasks)
    max_freq = max(task_count.values())
    max_freq_tasks = sum(1 for task in task_count if task_count[task] == max_freq)

    return max(len(tasks), (max_freq - 1) * (n + 1) + max_freq_tasks)
```

5. Number of 1 Bits (Apple)

Link

```
python
CopyEdit
def hammingWeight(n):
    count = 0
    while n:
        count += n & 1
        n >>= 1
    return count
```

6. Valid Parenthesis (Amazon, Facebook)

Link

```
python
CopyEdit
def isValid(s):
    stack = []
    mapping = {')': '(', '}': '{', ']': '['}

    for char in s:
        if char in mapping:
            top_element = stack.pop() if stack else '#'
            if mapping[char] != top_element:
                return False
        else:
            stack.append(char)

    return not stack
```

Day 23 (Miscellaneous - Hard)

7. Divide Two Integers (Adobe)

Link

python
CopyEdit

```
def divide(dividend, divisor):
    if dividend == -2**31 and divisor == -1:
        return 2**31 - 1

    sign = -1 if (dividend < 0) ^ (divisor < 0) else 1
    dividend, divisor = abs(dividend), abs(divisor)
    result = 0

    while dividend >= divisor:
        temp, multiple = divisor, 1
        while dividend >= temp << 1:
            temp <<= 1
            multiple <<= 1
        dividend -= temp
        result += multiple

    return sign * result
```

8. Trapping Rain Water (Apple)

Link

python
CopyEdit

```
def trap(height):
    if not height:
        return 0

    left, right = 0, len(height) - 1
    left_max, right_max = height[left], height[right]
    result = 0

    while left < right:
        if height[left] < height[right]:
            left += 1
            left_max = max(left_max, height[left])
            result += max(0, left_max - height[left])
        else:
            right -= 1
            right_max = max(right_max, height[right])
            result += max(0, right_max - height[right])

    return result
```

9. Max Number of Tasks You Can Assign (Apple, Nvidia)

Link

python
CopyEdit

```
def maxTasks(tasks, workers):
    tasks.sort()
    workers.sort()
    i, j = 0, 0
    count = 0
    while i < len(tasks) and j < len(workers):
        if workers[j] >= tasks[i]:
            count += 1
            i += 1
        j += 1

    return count
```

Day 24 (Miscellaneous - Hard)

10. Serialize and Deserialize Binary Tree (Amazon)

Link

```
python
CopyEdit
class Codec:

    def serialize(self, root):
        def dfs(node):
            if not node:
                return 'None'
            return str(node.val) + ',' + dfs(node.left) + ',' + dfs(node.right)

        return dfs(root)

    def deserialize(self, data):
        def dfs(data_list):
            if data_list[0] == 'None':
                data_list.pop(0)
                return None
            node = TreeNode(int(data_list.pop(0)))
            node.left = dfs(data_list)
            node.right = dfs(data_list)
            return node

        data_list = data.split(',')
        return dfs(data_list)
```

11. LRU Cache (Amazon)

Link

```
python
CopyEdit
from collections import OrderedDict

class LRUCache:

    def __init__(self, capacity: int):
        self.cache = OrderedDict()
        self.capacity = capacity

    def get(self, key: int) -> int:
        if key not in self.cache:
            return -1
        else:
            self.cache.move_to_end(key)
            return self.cache[key]
    def put(self, key: int, value: int) -> None:
        if key in self.cache:
            self.cache.move_to_end(key)
        self.cache[key] = value
        if len(self.cache) > self.capacity:
            self.cache.popitem(last=False)
```

12. Hamming Distance (Adobe)

Link

```
python
CopyEdit
def hammingDistance(x, y):
    return bin(x ^ y).count('1')
```