

39. Number of Ways to Wear Different Hats to Each Other There are n people and 40 types of hats labeled from 1 to 40. Given a 2D integer array `hats`, where `hats[i]` is a list of all hats preferred by the i th person. Return the number of ways that the n people wear different hats to each other. Since the answer may be too large, return it modulo $10^9 + 7$. Example 1: Input: `hats = [[3,4],[4,5],[5]]` Output: 1 Explanation: There is only one way to choose hats given the conditions. First person choose hat 3, Second person choose hat 4 and last one hat 5. Example 2: Input: `hats = [[3,5,1],[3,5]]` Output: 4 Explanation: There are 4 ways to choose hats: (3,5), (5,3), (1,3) and (1,5) Example 3: Input: `hats = [[1,2,3,4],[1,2,3,4],[1,2,3,4],[1,2,3,4]]` Output: 24 Explanation: Each person can choose hats labeled from 1 to 4. Number of Permutations of (1,2,3,4) = 24.

PROGRAM:

```
def numberWays(hats):
    MOD = 10**9 + 7
    n = len(hats)

    dp = [0] * (1 << n)
    dp[0] = 1 # Base case: No one wears a hat

    person_to_hats = {}
    for i, person_hats in enumerate(hats):
        for hat in person_hats:
            if hat not in person_to_hats:
                person_to_hats[hat] = []
            person_to_hats[hat].append(i)

    for hat in range(1, 41):
        new_dp = dp[:]

        for mask in range(1 << n):
            for person in person_to_hats.get(hat, []):
                if not (mask & (1 << person)):
                    new_dp[mask | (1 << person)] += dp[mask]
                    new_dp[mask | (1 << person)] %= MOD
```

```
dp = new_dp
```

```
return dp[(1 << n) - 1]
```

```
hats1 = [[3,4],[4,5],[5]]
```

```
print(numberWays(hats1))
```

```
hats2 = [[3,5,1],[3,5]]
```

```
print(numberWays(hats2))
```

```
hats3 = [[1,2,3,4],[1,2,3,4],[1,2,3,4],[1,2,3,4]]
```

```
print(numberWays(hats3))
```

```
1
4
24
```

OUTPUT:

TIME COMPLEXITY: $O(n \cdot 2^n \cdot m)$