

有n个无区别的物品，分成m个组，求划分方法模M的余数。

分析：

1. n的m划分，令 $dp[i][j]$ 表示j的i划分个数。
2. 当 $dp[i][j]$ 在 $j \geq i$ 时(即满足划分时)，可以表示为 $dp[i-1][j]$ (j的i-1划分)与 $dp[i][j-i]$ (j-i的i划分)之和。
3. 否则 $dp[i][j]$ 与 $dp[i-1][j]$ 相同。

```
class Solution{
public:
    int MNcount(int n,int m, int M) {
        std::vector<std::vector<int>> dp = std::vector<std::vector<int>>
(m+1, std::vector<int>(n+1,0));
        dp[0][0] = 1;
        for(int i = 1;i<=m;i++){
            for(int j = 0;j<=n;j++){
                if(j >= i) {
                    dp[i][j] = (dp[i-1][j] + dp[i][j-i]) % M;
                }else{
                    dp[i][j] = dp[i-1][j];
                }
            }
        }
    }
};
```

## 多重集组合数

有n种物品，第i种物品有 $a_i$ 个，不同物品可以区分，但相同物品无法区分。从这些物品中选取M个有多少种取法。

分析：设 $dp[i][j]$ 表示前i个物品取j个的组合总数。有 $dp[i][j] = dp[i][j-1] + dp[i-1][j] - dp[i-1][j-1-a_i]$

```
class Solution{
public:
    int multiNMcount(std::vector<int> &a, int n , int m) {
        std::vector<std::vector<int>> dp = std::vector<std::vector<int>>
(n+1, std::vector<int>(m+1,0));
        // base case 一个不取方法有一种
        for(int i =0;i<= n;i++) {
            dp[i][0] = 1;
        }
    }
};
```

```
    }

    for(int i=0;i<n;i++) {
        for(int j = 1;j <= m;j++){
            if(j-1-a[i] >= 0 ) {
                dp[i+1][j] = (dp[i+1][j] + dp[i][j] - dp[i][j-1-a[i]]
+ M)%M
            }else{
                dp[i+1][j] = (dp[i+1][j-1] + dp[i][j])%M;
            }
        }
    }
};
```