

{% note info %} **摘要** Title: 900. 整数划分 Tag: 计数类dp、完全背包 Memory Limit: 64 MB Time Limit: 1000 ms  
{% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

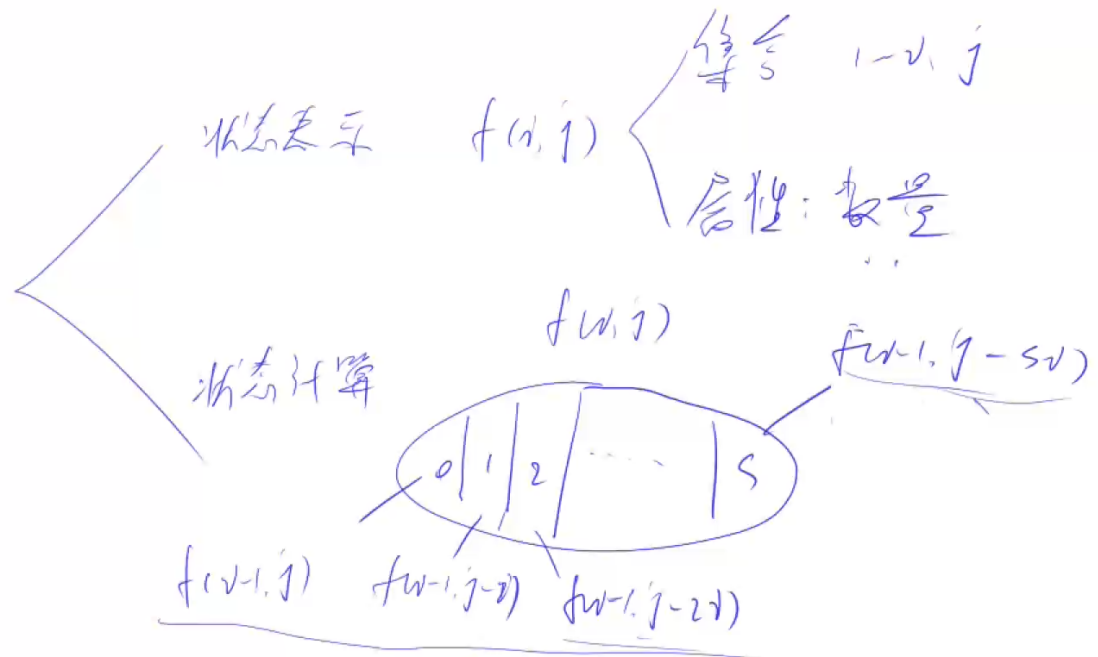
## 900. 整数划分

### • 题意

一个正整数  $n$  可以表示成若干个正整数之和，形如： $n = n_1 + n_2 + \dots + n_k$ ，其中  $n_1 \geq n_2 \geq \dots \geq n_k, k \geq 1$ 。我们将这样的一种表示称为正整数  $n$  的一种划分。现在给定一个正整数  $n$ ，请你求出  $n$  共有多少种不同的划分方法。

### • 思路

- **完全背包** 问题可以转化为：背包容量为  $n$ ，第  $i$  个物品的体积为  $i (i = 1 \sim n)$ ，每个物品有无限个，问恰好装满背包的**方案数**



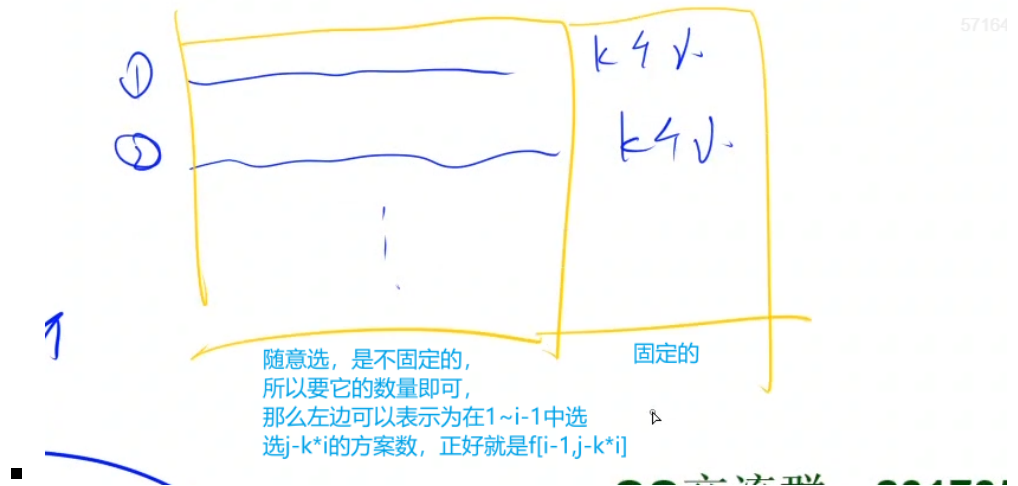
#### ■ 动态规划

##### ■ 状态表示

- 集合： $f[i][j]$ 表示只从  $1 \sim i$  物品中选，体积**恰好为** $j$ 的方案数
- 属性：数量

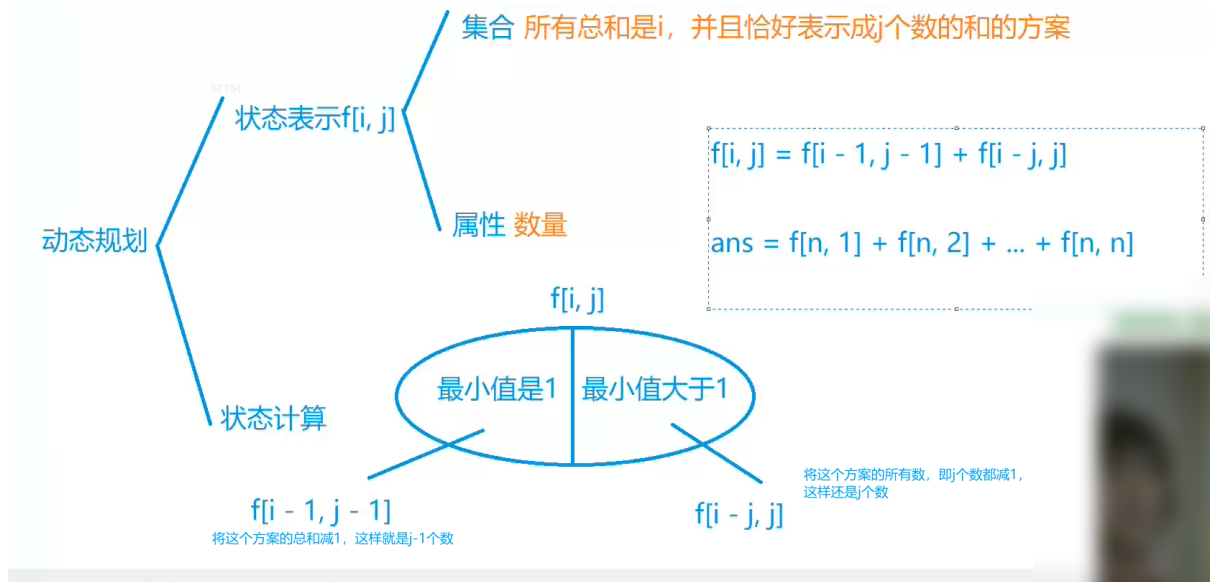
##### ■ 状态计算

- 看最后一个物品是**选几个**，即  $f[i-1, j]$  代表不选第  $i$  个物品， $f[i-1][j-k*i]$  代表选  $k$  个第  $i$  个物品，道理如下图



最后类似于背包问题, 从 $O(n^3)$ 优化为 $O(n^2)$ , 从二维优化为一维即可 最终优化为:  $dp[j] = (dp[j] + dp[j - i])$

### 计数dp



### 代码

#### 完全背包

```
...
Author: NEFU AB-IN
Date: 2022-03-07 21:21:51
FilePath: \ACM\Acwing\900.py
LastEditTime: 2022-03-07 21:24:45
...

N = 1010
MOD = int(1e9 + 7)
dp = [0] * N

n = int(input())
```

```

dp[0] = 1 #初始化: 代表一个数都不选时, 体积是0, 方案数是1

for i in range(1, n + 1):
    for j in range(i, n + 1): # (v[i], m + 1) -> (i, n + 1)
        dp[j] = (dp[j] + dp[j - i]) % MOD

print(dp[n])

```

### 计数dp

```

'''
Author: NEFU AB-IN
Date: 2022-03-07 22:28:25
FilePath: \ACM\Acwing\900.1.py
LastEditTime: 2022-03-07 22:31:58
'''

N = 1100
MOD = int(1e9 + 7)
dp = [[0] * N for _ in range(N)] # 表示总和为i, 并且恰好为j个数的方案

n = int(input())
dp[0][0] = 1 #总和为0, 恰好0个数的方案有一个

for i in range(1, n + 1):
    for j in range(1, n + 1):
        dp[i][j] = (dp[i - 1][j - 1] + dp[i - j][j]) % MOD

res = 0
for i in range(1, n + 1):
    res = (res + dp[n][i]) % MOD
print(res)

```