

The background is a faded anime-style illustration of a city street. In the center, a young man with dark, spiky hair and a light-colored trench coat stands looking towards a young woman with long, dark hair, wearing a brown jacket and a red scarf. They are standing in front of a large building with many windows. Below the windows, there is a yellow banner with Japanese text. At the bottom, there are silhouettes of other people walking on the street.

C4-L XIAO7和奇怪的排列

20376290-韩一

Description

- 对于排列 $\{a_i\}, \{b_i\}$, 定义: $MAX(a, b) = \sum_{i=1}^n \max\{a_i, b_i\}$
- 对于给定非负整数 K , 若满足 $MAX(a, b) \leq K$, 则认为是 (a, b) 是一对奇怪的排列。
- 给定 n 和 K , 求有多少对长度为 n 的、奇怪的排列。
- 答案对998244353取模。
- $n \leq 50, K \leq 2500$

Solution – 前置

- 一个序列是否“奇怪”与顺序无关，故我们指定 $b_i = i$ ，求此时 a_i 的方案数，将方案数乘以 $n!$ 即为答案。

Solution – DP状态表示

- 我们只考虑前 i 个位置里前 i 个数的填写情况
- 设 $f[i][j][k]$ 表示[前 i 个数]、[有 j 个位置待填充(同时有 j 个数未填)]、[已经确定的位置的 $\max\{a_i, i\}$ 的和为 k]的方案数
- 答案即为 $\sum_{i \leq K} f[n][0][i]$

Solution – DP状态转移

- 对于 $f[i][j][k]$, 我们考虑它能转移到哪些状态:
- (考虑第 $(i+1)$ 个位置填什么 + 第 $(i+1)$ 个数放在哪儿)
- 1. 第 $(i+1)$ 个位置空着, 第 $(i+1)$ 个数留到之后再填: $\sim \rightarrow f[i+1][j+1][k]$
- 2. 第 $(i+1)$ 个数填到第 $(i+1)$ 个位置: $\sim \rightarrow f[i+1][j][k+(i+1)]$
- 3. 第 $(i+1)$ 个数填到前面 j 个空位中的一个, 第 $(i+1)$ 个位置留到之后再填:
 - $j*\sim \rightarrow f[i+1][j][k+(i+1)]$
- 4. 前面 j 个没填的数中选一个填到第 $(i+1)$ 个位置, 第 $(i+1)$ 个数留到之后再填:
 - $j*\sim \rightarrow f[i+1][j][k+(i+1)]$
- 5. j 个没填的数中选一个填到第 $(i+1)$ 个位置, 第 $(i+1)$ 个数填到填到 j 个空位中的一个:
 - $j*j*\sim \rightarrow f[i+1][j-1][k+2(i+1)]$
- 初始状态为 $f[0][0][0]=1$, 直接转移即可, 时间复杂度 $O(n^2K)$

Example

- 前面四个位置的情况是2,(),3,(), 即位置2,4待填, 数字1,4待填, 属于f[4][2][5]
- 考虑第五个位置和第五个数的填写情况
- 1. 2,(),3,(),(), 即f[5][3][5]
- 2. 2,(),3,(),5, 即f[5][2][10]
- 3. 2,5,3,(),(), 即f[5][2][10]
- 4. 2,(),3,(),4, 即f[5][2][10]
- 5. 2,5,3,(),1, 即f[5][1][15]



Code

```
n = read<int>(), K = read<int>();

//f[i][j][k]: for the first i elements, the number of blank space is j, and the sum of definite positions is k
for (int i = 0; i <= K; ++ i) f[0][0][i] = 1;
for (int i = 0; i < n; ++ i) for (int j = 0; j <= i; ++ j) for (int k = 0; k <= K; ++ k) {
    (f[i + 1][j + 1][k] += f[i][j][k]) %= MOD;
    if (k + i + 1 <= K) f[i + 1][j][k + (i + 1)] = (f[i + 1][j][k + (i + 1)] + (LL)f[i][j][k] * (j * 2 + 1)) % MOD;
    if (j && k + 2 * (i + 1) <= K) f[i + 1][j - 1][k + 2 * (i + 1)] = (f[i + 1][j - 1][k + 2 * (i + 1)] + (LL)f[i][j][k] * j * j) % MOD;
}
int ans = f[n][0][K];
for (int i = 2; i <= n; ++ i)
    ans = (LL)ans * i % MOD;
printf("%d\n", ans);
```

THANKS

