Rozdělení

Pro permutaci $p = p[0]\,p[1]\,p[2]\,\ldots\,p[n-1]$ čísel $1, 2, 3, \ldots, n$ definujeme *rozdělení* jako permutaci q, kterou lze získat následujícím procesem:

1. Vezmeme dvě množiny čísel $A = \{\, i_1, i_2, ..., i_k \,\}$ a $B = \{\, j_1, j_2, ..., j_l \,\}$ takové, že $A \cap B = \emptyset$, $A \cup B = \{\, 0, 1, 2, ..., n-1 \,\}$, $i_1 < i_2 < ... < i_k$ a $j_1 < j_2 < ... < j_l$.
2. Permutaci q určíme jako $q = p[i_1]\,p[i_2]\,\ldots\,p[i_k]\,p[j_1]\,p[j_2]\,\ldots\,p[j_l]$.

Dále definujeme $S(p)$ jako množinu všech rozdělení permutace p.

Dostanete číslo n a množinu T obsahující m permutací délky n. Spočítejte, kolik existuje permutací p délky n takových, že $T \subseteq S(p)$. Protože toto číslo může být velké, najděte jeho zbytek po dělení číslem $998\,244\,353$.

Implementační podrobnosti

Máte implementovat následující funkci:

```
int solve(int n, int m, std::vector<std::vector<int>>& splits);
```

- n je velikost permutace.
- m je počet rozdělení.
- $splits$ je pole m **po dvou různých** permutací, prvků množiny T, která má být podmnožinou $S(p)$.
- Funkce má vrátit počet možných permutací p modulo $998\,244\,353$.
- Tato funkce bude zavolána právě jednou za každé spuštění programu.

Omezení

- $1 \leq n \leq 300$
- $1 \leq m \leq 300$

Podúlohy

1. (6 bodů) $m = 1$
2. (7 bodů) $1 \leq n, m \leq 10$
3. (17 bodů) $1 \leq n, m \leq 18$
4. (17 bodů) $1 \leq n \leq 30, 1 \leq m \leq 15$
5. (16 bodů) $1 \leq n, m \leq 90$
6. (16 bodů) $1 \leq n \leq 300$, $1 \leq m \leq 15$
7. (21 bodů) *Bez dalších omezení.*

Ukázka

Uvažujme následující volání:

```
solve(3, 2, {{1, 2, 3}, {2, 1, 3}})
```

V této ukázce je velikost permutace p rovná 3 a máme zadaná 2 rozdělení:

- $1\ 2\ 3$
- $2\ 1\ 3$

Funkce vrátí 4, protože existují pouze čtyři permutace p, které mohou vygenerovat obě zadaná rozdělení:

- $1\ 2\ 3$
- $1\ 3\ 2$
- $2\ 1\ 3$
- $2\ 3\ 1$

Ukázkový grader

Ukázkový grader čte vstup v následujícím formátu:

- řádek 1: $n\ m$
- řádek $2 + i$: $s[i][0]\ s[i][1]\ \ldots\ s[i][n-1]$ pro všechna $0 \leq i < m$

a vypíše návratovou hodnotu volání `solve` s odpovídajícími parametry.