

Split

Sei $p = p[0], p[1], p[2], \dots, p[n-1]$ eine Permutation der Zahlen $1, 2, 3, \dots, n$. Wir sagen, dass eine Permutation q ein *Split von p* ist, falls wir q aus p wie folgt erhalten können:

1. Wähle $A = \{i_1, i_2, \dots, i_k\}$ und $B = \{j_1, j_2, \dots, j_l\}$ so, dass $A \cap B = \emptyset$, $A \cup B = \{0, 1, 2, \dots, n-1\}$, $i_1 < i_2 < \dots < i_k$ und $j_1 < j_2 < \dots < j_l$.
2. Definiere q als $q = p[i_1], p[i_2], \dots, p[i_k], p[j_1], p[j_2], \dots, p[j_l]$.

Definiere $S(p)$ als die Menge aller möglichen *Splits* der Permutation p .

In dieser Aufgabe hast du eine Zahl n und eine Menge T mit m Permutationen der Länge n gegeben. Zähle, wie viele Permutationen p mit Länge n es gibt, für die $T \subseteq S(p)$ gilt (das Symbol \subseteq bedeutet Teilmenge, d.h. wir zählen die Anzahl p , für die T eine Teilmenge von $S(p)$ ist). Da diese Anzahl sehr gross werden kann, gib sie modulo 998 244 353 zurück.

Implementierungsdetails

Implementiere die folgende Funktion:

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

- n : Die Länge der Permutationen
- m : Die Anzahl der Splits
- *splits*: ein Array mit m **paarweise verschiedenen** Permutationen, den Elementen der Menge T , das eine Teilmenge von $S(p)$ ist.
- Die Funktion soll die Anzahl an möglichen Permutationen p modulo 998 244 353 zurückgeben.
- Die Funktion wird genau einmal pro Testfall aufgerufen.

Beschränkungen

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

Teilaufgaben

1. (6 Punkte) $m = 1$
2. (7 Punkte) $1 \leq n, m \leq 10$

3. (17 Punkte) $1 \leq n, m \leq 18$
4. (17 Punkte) $1 \leq n \leq 30, 1 \leq m \leq 15$
5. (16 Punkte) $1 \leq n, m \leq 90$
6. (16 Punkte) $1 \leq n \leq 300, 1 \leq m \leq 15$
7. (21 Punkte) Keine weiteren Einschränkungen.

Beispiele

Beispiel 1

Betrachte den folgenden Aufruf:

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

In diesem Beispiel hat die Permutation p die Länge 3 und es sind 2 Splits gegeben:

- 1 2 3
- 2 1 3

Der Funktionsaufruf soll 4 zurückgeben, da es genau vier Möglichkeiten für p gibt, mit denen es möglich ist, beide Splits zu generieren:

- 1 2 3
- 1 3 2
- 2 1 3
- 2 3 1

Beispiel-Grader

Der Beispiel-Grader liest die Eingabe in folgendem Format:

- Zeile 1: $n \ m$
- Zeile $2 + i$: $s[i][0] \ s[i][1] \ \dots \ s[i][n - 1]$ für alle $0 \leq i < m$

und gibt den Rückgabewert von `solve` mit den entsprechenden Parametern aus.