

Splits

Für eine Permutation $p = p[0] p[1] p[2] \dots p[n-1]$ der Zahlen $1, 2, 3, \dots, n$ definieren wir einen *Split* als eine Permutation q , die man durch den folgenden Prozess erhält:

1. Wähle zwei Mengen an Zahlen $A = i_1, i_2, \dots, i_k$ und $B = j_1, j_2, \dots, j_l$, sodass $A \cap B = \emptyset$, $A \cup B = 0, 1, 2, \dots, n-1$, $i_1 < i_2 < \dots < i_k$ und $j_1 < j_2 < \dots < j_l$
2. Definiere q als $q = p[i_1]p[i_2] \dots p[i_k]p[j_1]p[j_2] \dots p[j_l]$

Außerdem wird die Menge aller *Splits* einer Permutation p mit $S(p)$ bezeichnet.

Du erhältst eine ganze Zahl n und eine Menge T von m Permutationen der Länge n . Bestimme, wie viele Permutationen p der Länge n existieren, für die $T \subseteq S(p)$ gilt. Da diese Zahl sehr groß werden kann, sollst du sie modulo 998 244 353 ausgeben.

Implementierungshinweise

Implementiere die folgende Funktion

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

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

Beschränkungen

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

Teilaufgaben

1. (7 Punkte) $1 \leq n, m \leq 10$
2. (17 Punkte) $1 \leq n, m \leq 18$

3. (19 Punkte) $1 \leq n \leq 30, 1 \leq m \leq 15$
4. (18 Punkte) $1 \leq n, m \leq 90$
5. (16 Punkte) $1 \leq n \leq 300, 1 \leq m \leq 15$
6. (23 Punkte) Keine weiteren Beschränkungen.

Beispiele

Beispiel 1

Betrachte den folgenden Funktionsaufruf

```
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 gibt 4 zurück, da es nur vier mögliche Permutationen gibt, die beide Splits erzeugen:

- 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 das Ergebnis des Aufrufs von `solve` mit den entsprechenden Parametern aus.