

## Cięcia

Dla permutacji  $p = p[0] p[1] p[2] \dots p[n-1]$  liczb  $1, 2, 3, \dots, n$ , definiujemy *cięcie* jako permutację  $q$ , którą można otrzymać w wyniku następującego procesu:

1. Wybierz dwa zbiory liczb  $A = \{i_1, i_2, \dots, i_k\}$  oraz  $B = \{j_1, j_2, \dots, j_l\}$  takie, że  $A \cap B = \emptyset$ ,  $A \cup B = \{0, 1, 2, \dots, n-1\}$ ,  $i_1 < i_2 < \dots < i_k$  a także  $j_1 < j_2 < \dots < j_l$
2. Permutacją  $q$  będzie wtedy  $q = p[i_1]p[i_2] \dots p[i_k]p[j_1]p[j_2] \dots p[j_l]$

Co więcej, oznaczmy przez  $S(p)$  zbiór wszystkich *cięć* permutacji  $p$ .

Dana jest liczba  $n$  oraz zbiór  $T$  złożony z  $m$  permutacji rozmiaru  $n$ . Wyznacz, jak wiele istnieje permutacji  $p$  rozmiaru  $n$  takich, że  $T \subseteq S(p)$ . Ta liczba może być ogromna, więc policz ją modulo 998 244 353.

## Szczegóły implementacji

Zaimplementuj następującą funkcję:

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

- $n$ : rozmiar permutacji
- $m$ : liczba danych cięć
- *splits*: wektor zawierający  $m$  **parami różnych** permutacji, tj. elementów zbioru  $T$ , który jest podzbiorem  $S(p)$
- Ta funkcja powinna zwrócić liczbę wszystkich możliwych permutacji modulo 998 244 353.
- Ta funkcja będzie wywołana dokładnie raz dla każdego testu.

## Ograniczenia

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

## Podzadania

1. (6 punktów)  $m = 1$
2. (7 punktów)  $1 \leq n, m \leq 10$
3. (17 punktów)  $1 \leq n, m \leq 18$

4. (17 punktów)  $1 \leq n \leq 30, 1 \leq m \leq 15$
5. (16 punktów)  $1 \leq n, m \leq 90$
6. (16 punktów)  $1 \leq n \leq 300, 1 \leq m \leq 15$
7. (21 punktów) Brak dodatkowych ograniczeń.

## Przykłady

### Przykład 1

Rozważmy następujące wywołanie:

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

W tym przykładzie, rozmiarem permutacji  $p$  jest 3 i mamy dane 2 cięcia:

- 1 2 3
- 2 1 3

Wywołanie to powinno zwrócić 4 jako, że są tylko cztery możliwe warianty  $p$ , które generują obydwie te cięcia:

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

## Przykładowa sprawdzaczka

Przykładowa sprawdzaczka odczytuje dane w następującym formacie:

- linia 1:  $n \ m$
- linia  $2 + i$ :  $s[i][0] \ s[i][1] \ \dots \ s[i][n-1]$  dla każdego  $0 \leq i < m$

i wypisuje wynik wywołania `solve` z zadanymi parametrami.