Виставка Настільних Ігор

Щороку в Клуж-Напоці відбувається велика Виставка Настільних Ігор, на якій представлено широкий вибір нових ігор. Головною родзинкою цього року є гра під назвою BoardOina.

У черзі на гру стоїть n гравців. Гравці пронумеровані від 0 до $n-1$ у порядку черги. Гравець 0 стоїть першим, а гравець $n-1$ — останнім.

Існує m *різних* **дружніх зв'язків** між m парами гравців у черзі. Зокрема, для кожного i від 0 до $m-1$ включно, гравці $x[i]$ і $y[i]$ є друзями, де $0 \leq x[i] < y[i] < n$. Дружба є взаємною.

Розглянемо послідовність з k *послідовних* гравців у черзі, починаючи з гравця s (для будь-яких s і k, таких що $0 \leq s < n$ і $1 \leq k \leq n-s$). Ця послідовність утворює **дружню групу** розміру k, якщо для будь-якої пари гравців серед них існує послідовність дружніх зв'язків у межах цієї групи. Зокрема, гравці $s, s+1, \ldots, s+k-1$ утворюють дружню групу розміру k, якщо для кожних u і v таких, що $s \leq u < v < s+k$, існує послідовність гравців $p[0], \ldots, p[l-1]$ така, що:

- $l \geq 2$;
- $s \leq p[j] < s+k$ для кожного j від 0 до $l-1$ включно;
- $p[0] = u$ і $p[l-1] = v$;
- гравці $p[j]$ і $p[j+1]$ є друзями для кожного j від 0 до $l-2$ включно.

Зверніть увагу: у випадку $k = 1$ гравець s сам утворює дружню групу розміру 1.

У BoardOina може грати будь-яка кількість гравців. Однак, щоб гра виглядала цікавіше, організатори дозволяють грати лише дружнім групам.

Одночасно може грати лише одна група. Для кожної гри формується дружня група, починаючи з гравця на початку черги, і вона одразу починає гру. Ці гравці видаляються з черги. Процес повторюється, поки черга не спорожніє. Формально, чергу **можна розбити на** g **дружніх груп**, якщо існує масив розмірів груп $K = [K[0], K[1], \ldots, K[g-1]]$, такий що виконуються всі умови:

- $g > 0$ та $K[j] > 0$ для кожного j такого, що $0 \leq j < g$;
- $K[0] + K[1] + \ldots + K[g-1] = n$;
- для кожного j від 0 до $g-1$ включно, гравці $s[j], s[j]+1, \ldots, s[j]+K[j]-1$ утворюють дружню групу розміру $K[j]$, де $s[0] = 0$, а далі $s[j] = K[0] + K[1] + \ldots + K[j-1]$.

Організатори хочуть *мінімізувати* кількість дружніх груп, які гратимуть у гру. Тобто, вони прагнуть розбити чергу на g дружніх груп так, щоб неможливо було зробити це з меншою кількістю груп.

Ваше завдання — знайти розбиття черги на мінімальну кількість дружніх груп і вивести масив розмірів груп.

Деталі реалізації

Вам слід реалізувати наступну процедуру:

```
std::vector<int> partition_players(int n, int m, std::vector<int> X, std::vector<int> Y)
```

- n: кількість гравців у черзі.
- m: кількість дружніх зв'язків.
- x, y: масиви довжини m, які описують дружні зв'язки.
- Ця процедура має повернути масив розмірів груп, що задає розбиття черги на мінімальну кількість дружніх груп.
- Ця процедура викликається рівно один раз для кожного тесту.

Обмеження

- $2 \leq n \leq 100\,000$
- $0 \leq m \leq 200\,000$
- $0 \leq x[i] < y[i] < n$ для кожного i такого, що $0 \leq i < m$
- Усі дружні зв'язки різні. Інакше кажучи, $x[i] \neq x[j]$ або $y[i] \neq y[j]$ для кожного i і j, таких що $0 \leq i < j < m$.
- Якщо існує декілька коректних розбиттів з мінімальною кількістю груп, можна повернути будь-яке з них.

Підзадачі

1. (5 балів) $y[i] = x[i] + 1$ для кожного i від 0 до $m - 1$ включно.
2. (7 балів) $y[i] \leq x[i] + 2$ для кожного i від 0 до $m - 1$ включно.
3. (6 балів) $n \leq 300$ і $m \leq 600$
4. (15 балів) $n \leq 2\,000$ і $m \leq 4\,000$
5. (34 бали) Відсутні дружні зв'язки, які утворюють *цикл*. Тобто, для будь-якої послідовності *різних* гравців $p[0], p[1], \ldots, p[l-1]$, де $l \geq 3$ і для кожного $0 \leq j < l - 1$ гравці $p[j]$ і $p[j+1]$ є друзями, гравці $p[0]$ і $p[l-1]$ **не** є друзями.
6. (33 бали) Без додаткових обмежень.

Приклади

Приклад 1

Розглянемо виклик:

```
partition_players(5, 3, {0, 1, 3}, {1, 4, 4})
```

У цьому прикладі гравці 0 і 1, 1 і 4, а також 3 і 4 є друзями.

Гравець 2 не має друзів у черзі, тому мусить бути окрема група лише з гравцем 2, тож мінімальна кількість груп $g = 3$. З іншого боку, 0 і 1, а також 3 і 4 можуть бути групами по двоє.

Отже, чергу можна розбити на 3 групи розмірів 2, 1 і 2, тому процедура може повернути $[2, 1, 2]$.

Приклад 2

Розглянемо виклик:

```
partition_players(7, 6, {0, 4, 2, 1, 2, 3}, {1, 5, 4, 5, 5, 6})
```

У цьому прикладі гравці 0 і 1, 4 і 5, 2 і 4, 1 і 5, 2 і 5, 3 і 6 є друзями.

Єдиний друг гравця 3 — це гравець 6, отже будь-яка група з гравцем 3 — це або:

- група розміром 1, яка містить тільки 3, або
- група з 3 та 6.

У другому випадку група мала б також містити 4 і 5, але це неможливо, бо єдиний друг 6 — це 3, тож 3 не з'єднаний із 4 та 5 послідовністю дружніх зв'язків.

Отже, 3 повинен бути в окремій групі розміром 1. Аналогічно, 6 також повинен бути в окремій групі, тож мінімум 4 групи.

Гравці 0, 1 і 2 не можуть утворити групу з 3 осіб, оскільки 0 і 1 не зв'язані з 2 послідовністю дружніх зв'язків в межах цієї групи. Це було б можливо, якби 5-й гравець також був у групі, але оскільки 3 і 4 гарантовано будуть в різних групах, то це неможливо. Отже, мінімум 5 груп.

З іншого боку, 0 і 1, а також 4 і 5 утворюють дві дружні групи розміром 2. Отже, чергу можна розбити на 5 груп розмірів 2, 1, 1, 2 і 1. Процедура може повернути $[2, 1, 1, 2, 1]$.

Приклад градера

Зчитує вхідні дані у наступному форматі:

- рядок 1: $n \ m$
- рядки $2 + i$ ($0 \leq i < m$): $x[i] \ y[i]$

Нехай елементи масиву, який повертає `partition_players` це $K[0], K[1], \ldots, K[g-1]$ для деякого додатного g. Вивід градера має такий формат:

- рядок 1: g
- рядок 2: $K[0] \ K[1] \ \ldots \ K[g-1]$