Brettspiel-Expo

Jedes Jahr findet in Cluj-Napoca eine grosse Brettspiel-Expo statt, auf der eine breite Auswahl neuer Spiele vorgestellt wird. Die Hauptattraktion in diesem Jahr ist ein Spiel namens BoardOina.

Es stehen n Spieler in einer Warteschlange, die darauf warten, das Spiel auszuprobieren. Die Spieler sind von 0 bis $n-1$ durchnummeriert, entsprechend ihrer Position in der Schlange. Spieler 0 steht ganz vorne in der Schlange, Spieler $n-1$ ganz hinten.

Es gibt m *verschiedene* **Freundschaftsbeziehungen** zwischen m Spielerpaaren in der Schlange. Genauer gesagt sind für jedes i von 0 bis $m-1$ (inklusive) die Spieler $x[i]$ und $y[i]$ befreundet, wobei $0 \leq x[i] < y[i] < n$ gilt. Freundschaftsbeziehungen sind symmetrisch.

Betrachte eine Sequenz von k *aufeinanderfolgenden* Spielern in der Schlange, beginnend bei Spieler s (für beliebige s und k, für die $0 \leq s < n$ und $1 \leq k \leq n-s$ gilt). Diese Spieler bilden eine **Freundesgruppe** der Grösse k, wenn für alle Paare von zwei Spielern eine Reihe von Freundschaftsbeziehungen innerhalb dieser Gruppe existiert, die sie verbindet. Genauer gesagt bilden die Spieler $s, s+1, \ldots, s+k-1$ eine Freundesgruppe der Grösse k, wenn es für jedes u und v mit $s \leq u < v < s+k$ eine Spielerfolge $p[0], \ldots, p[l-1]$ gibt, so dass:

- $\ell \geq 2$;
- $s \leq p[j] < s+k$ für jedes j von 0 bis $\ell-1$ (inklusive);
- $p[0] = u$ und $p[\ell-1] = v$;
- Spieler $p[j]$ und $p[j+1]$ sind befreundet für jedes j von 0 bis $\ell-2$ (inklusive).

Beachte, dass im Fall $k=1$ der Spieler s allein eine Freundesgruppe der Grösse 1 bildet.

BoardOina kann von beliebig vielen Spielern gespielt werden. Damit das Spiel erfolgreicher wird, lassen die Veranstalter jedoch nur Freundesgruppen spielen.

Jeweils nur eine Gruppe kann gleichzeitig spielen. Für jedes Spiel wird eine Freundesgruppe gebildet, beginnend mit dem Spieler an der Spitze der Warteschlange, die dann das Spiel spielt. Die Spieler dieser Freundesgruppe werden aus der Schlange entfernt. Dieser Vorgang wird wiederholt, bis die Schlange leer ist. Formal sagen wir, dass die Schlange **in g Freundesgruppen aufgeteilt werden kann**, wenn es ein Array von Gruppengrössen $K = [K[0], K[1], \ldots, K[g-1]]$ gibt, so dass jede der folgenden Bedingungen erfüllt ist:

- $g > 0$ und $K[j] > 0$ (für jedes j mit $0 \leq j < g$);
- $K[0] + K[1] + \ldots + K[g-1] = n$;

- für jedes j von 0 bis $g-1$ (inklusive) bilden die Spieler $s[j], s[j+1], \ldots, s[j+K[j]-1]$ eine Freundesgruppe der Grösse $K[j]$, wobei $s[0]=0$ und sonst $s[j]=K[0]+K[1]+\ldots+K[j-1]$.

Die Veranstalter möchten die Anzahl der spielenden Freundesgruppen *minimieren*. Das heisst, sie wollen die Schlange in g Freundesgruppen aufteilen, so dass eine Aufteilung in $g-1$ (oder weniger) Gruppen nicht möglich ist.

Deine Aufgabe ist es, eine Aufteilung der Schlange in eine minimale Anzahl von Freundesgruppen zu finden und das Array der Gruppengrössen zurückzugeben.

Implementierungsdetails

Du sollst folgende Funktion implementieren:

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

- n: Anzahl der Spieler in der Schlange.
- m: Anzahl der Freundschaftsbeziehungen.
- x, y: Ein `vector` der Länge m, die die Freundschaftsbeziehungen beschreiben.
- Diese Funktion soll ein `vector` von Gruppengrössen zurückgeben, das eine Aufteilung der Spielerschlange in eine minimale Anzahl von Freundesgruppen beschreibt.
- Diese Funktion wird für jeden Testfall genau einmal aufgerufen.

Einschränkungen

- $2 \leq n \leq 100\,000$
- $0 \leq m \leq 200\,000$
- $0 \leq x[i] < y[i] < n$ (für jedes i mit $0 \leq i < m$)
- Freundschaftsbeziehungen sind unterschiedlich. Das heisst, $x[i] \neq x[j]$ oder $y[i] \neq y[j]$ (für alle i und j mit $0 \leq i < j < m$).
- Wenn es mehrere Lösungen mit der minimalen Anzahl an Gruppen gibt, kannst du eine beliebige gültige Lösung zurückgeben.

Teilaufgaben

1. (5 Punkte) $y[i]=x[i]+1$ für jedes i von 0 bis $m-1$ (inklusive)
2. (7 Punkte) $y[i] \leq x[i]+2$ für jedes i von 0 bis $m-1$ (inklusive)
3. (6 Punkte) $n \leq 300$ und $m \leq 600$
4. (15 Punkte) $n \leq 2\,000$ und $m \leq 4\,000$
5. (34 Punkte) Es gibt keine zyklischen Freundschaftsbeziehungen. Das heisst, für jede Folge von *verschiedenen* Spielern $p[0], p[1], \ldots, p[l-1]$, mit $\ell \geq 3$ und für jedes $0 \leq j < \ell - 1$ sind

die Spieler $p[j]$ und $p[j+1]$ befreundet, aber die Spieler $p[0]$ und $p[\ell-1]$ **sind nicht** befreundet.
6. (33 Punkte) Keine weiteren Einschränkungen.

Beispiele

Beispiel 1

Betrachte den folgenden Aufruf:

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

In diesem Beispiel sind die Spieler 0 und 1, die Spieler 1 und 4 sowie die Spieler 3 und 4 befreundet.

Spieler 2 hat keine Freunde in der Schlange, daher muss eine Freundesgruppe gebildet werden, die nur aus Spieler 2 besteht, was bedeutet, dass die minimale Anzahl an Freundesgruppen $g=3$ ist. Andererseits können Spieler 0 und 1 sowie Spieler 3 und 4 jeweils eine Freundesgruppe der Grösse 2 bilden.

Daher kann die Schlange in 3 Freundesgruppen der Grössen 2, 1 und 2 aufgeteilt werden, sodass die Funktion $[2,1,2]$ zurückgeben kann.

Beispiel 2

Betrachte den folgenden Aufruf:

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

In diesem Beispiel sind die Spieler 0 und 1, 4 und 5, 2 und 4, 1 und 5, 2 und 5 sowie 3 und 6 befreundet.

Der einzige Freund von Spieler 3 ist Spieler 6, also kann eine Freundesgruppe, die Spieler 3 enthält, entweder

- eine Freundesgruppe der Grösse 1 sein, die nur Spieler 3 enthält, oder
- eine Freundesgruppe, die sowohl Spieler 3 als auch Spieler 6 enthält.

Eine Freundesgruppe im zweiten Fall müsste auch die Spieler 4 und 5 enthalten. Das ist jedoch nicht möglich, da der einzige Freund von Spieler 6 Spieler 3 ist, und Spieler 3 somit nicht über eine Freundschaftsbeziehung mit Spielern 4 und 5 verbunden ist.

Daher muss Spieler 3 in eine Freundesgruppe der Grösse 1 eingeteilt werden. Ebenso muss Spieler 6 in eine Freundesgruppe der Grösse 1 eingeteilt werden, daher ist die Anzahl der

Freundesgruppen in einer Partition mindestens 4.

Die Spieler 0, 1 und 2 bilden keine Freundesgruppe der Grösse 3, da weder Spieler 0 noch 1 mit Spieler 2 über eine Kette von Freundschaftsbeziehungen innerhalb der Gruppe verbunden sind. Das wäre nicht der Fall, wenn 5 in der Gruppe wäre. Dies kann aber nicht sein, da 3 und 4 definitiv in unterschiedlichen Gruppen sein müssen. Das heisst, die Anzahl der Freundesgruppen in einer Partition ist mindestens 5.

Andererseits bilden die Spieler 0 und 1 sowie die Spieler 4 und 5 zwei Freundesgruppen der Grösse 2. Daher kann die Schlange in 5 Freundesgruppen der Grössen 2, 1, 1, 2 und 1 aufgeteilt werden. Die Funktion kann $[2, 1, 1, 2, 1]$ zurückgeben.

Beispiel-Grader

Der Beispiel-Grader liest den Input in folgendem Format:

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

Seien die Elemente des `vector`, welches `partition_players` zurückgibt, $K[0], K[1], \ldots, K[g-1]$ für ein gewisses nichtnegatives g. Die Ausgabe des Beispiel-Grader erfolgt in folgendem Format:

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