Targi gier planszowych

Każdego roku, w Klużu odbywają się wielkie targi gier planszowych z prezentacjami szerokiego spektrum nowych gier. W tym roku główną atrakcją targów jest gra o nazwie BoardOina.

W targach bierze udział n graczy ustawionych w kolejce w oczekiwaniu na możliwość wypróbowania tej gry. Gracze są ponumerowani od 0 do $n-1$ w porządku, w jakim stoją w tej kolejce. Gracz 0 jest na jej czele a gracz $n-1$ stoi na jej końcu.

Pomiędzy parami graczy w kolejce zachodzi m *różnych* **związków przyjaźni**. Dokładniej, dla każdego i od 0 do $m-1$ włącznie, gracz $x[i]$ oraz gracz $y[i]$ są przyjaciółmi, przy czym $0 \leq x[i] < y[i] < n$. Relacja przyjaźni jest symetryczna.

Rozważmy ciąg k *kolejnych* graczy w kolejce poczynając od gracza s (dla dowolnego s oraz k takich, że $0 \leq s < n$ i $1 \leq k \leq n-s$). Taki ciąg graczy stanowi **sitwę** wielkości k, jeśli każda para członków tej grupy jest połączona ciągiem związków przyjaźni wewnątrz grupy. Mówiąc bardziej precyzyjnie, gracze $s, s+1, \ldots, s+k-1$ stanowią sitwę wielkości k, jeśli dla każdych u oraz v takich, że $s \leq u < v < s+k$, istnieje ciąg graczy $p[0], \ldots, p[l-1]$ taki, że:

- $l \geq 2$;
- $s \leq p[j] < s+k$ dla każdego j od 0 do $l-1$ włącznie;
- $p[0] = u$ oraz $p[l-1] = v$;
- gracze $p[j]$ oraz $p[j+1]$ są przyjaciółmi dla każdego j od 0 do $l-2$ włącznie.

Zauważ, że gdy $k=1$, sam gracz s stanowi sitwę rozmiaru 1.

W BoardOina może grać dowolnie wielu graczy, jednak by uczynić ją ciekawszą, organizatorzy pozwają grać w nią tylko sitwom.

Jednocześnie może grać tylko jedna sitwa. Grę zawsze rozpoczyna jakaś sitwa złożona z gracza znajdującego się na czele kolejki. Gracze należący do tej sitwy opuszczają kolejkę. Procedura ta jest powtarzana aż do chwili, gdy kolejka się opróżni. Formalnie, powiemy, że kolejkę **można podzielić na** g **sitw** jeśli istnieje tablica rozmiarów sitw $K = [K[0], K[1], \ldots, K[g-1]]$, taka że zachodzą jednocześnie wszystkie poniższe warunki.

- $g > 0$ oraz $K[j] > 0$ (dla każdego j takiego że $0 \leq j < g$);
- $K[0] + K[1] + \ldots + K[g-1] = n$;
- dla każdego j między 0 i $g-1$ włącznie, gracze $s[j], s[j]+1, \ldots, s[j]+K[j]-1$ stanowią sitwę rozmiaru $K[j]$, przy czym $s[0] = 0$ oraz $s[j] = K[0] + K[1] + \ldots + K[j-1]$ dla $j > 0$.

Organizatorzy chcą *zminimalizować* liczbę sitw, które grają w tę grę. Chcą więc podzielić tę kolejkę na g sitw tak, aby nie można jej było podzielić na $g-1$ lub mniej sitw.

Twoim zadaniem jest znalezienie podziału kolejki na minimalną liczbę sitw i przedstawić tablicę wielkości sitw.

Szczegóły implementacji

Zaimplementuj następującą procedurę:

```
std::vector< int> partition_players(int n, int m, std::vector< int> x, std::vector< int> y)
```

- n: liczba graczy w kolejce.

- m: liczba związków przyjaźni.
- x, y: wektory rozmiaru m z opisem związków przyjaźni.
- Ta procedura powinna zwrócić wektor rozmiarów sitw reprezentującą podział kolejki graczy na minimalną liczbę sitw.
- Ta procedura będzie wywoływana dokładnie raz dla każdego testu.

Ograniczenia

- $2 \leq n \leq 100\,000$
- $0 \leq m \leq 200\,000$
- $0 \leq x[i] < y[i] < n$ (dla każdego i takiego, że $0 \leq i < m$)
- Związki przyjaźni nie powtarzają się. Innymi słowy, $x[i] \neq x[j]$ lub $y[i] \neq y[j]$ dla każdych i oraz j takich, że $0 \leq i < j < m$.
- Jeśli istnieje wiele możliwych rozwiązań osiągających minimalną liczbę grup, możesz wypisać dowolne z nich.

Podzadania

1. (5 punktów) $y[i] = x[i] + 1$ dla każdego i od 0 do $m - 1$ włącznie.
2. (7 punktów) $y[i] \leq x[i] + 2$ dla każdego i od 0 do $m - 1$ włącznie.
3. (6 punktów) $n \leq 300$ i $m \leq 600$
4. (15 punktów) $n \leq 2\,000$ i $m \leq 4\,000$
5. (34 punktów) Nie ma związków przyjaźni, które są *cykliczne*. To znaczy, że dla dowolnego ciągu *różnych* graczy $p[0], p[1], \ldots, p[l-1]$, takich że $l \geq 3$ oraz dla każdego $0 \leq j < l - 1$ gracze $p[j]$ i $p[j+1]$ są przyjaciółmi, gracze $p[0]$ i $p[l-1]$ **nie** są przyjaciółmi.
6. (33 punktów) Bez dodatkowych ograniczeń.

Przykłady

Przykład 1

Rozważmy następujące wywołanie:

```
partition_players(5, 3, [0, 1, 3], [1, 4, 4])
```

W tym przykładzie gracze 0 i 1, gracze 1 and 4, oraz gracze 3 i 4 są przyjaciółmi.

Gracz 2 nie ma przyjaciół w kolejce, więc musi sam stanowić sitwę, co oznacza, że minimalna liczba sitw wynosi $g = 3$. Z drugiej strony gracze 0 i 1, a także gracze 3 i 4 stanowią sitwy rozmiaru 2.

Kolejkę można zatem podzielić na 3 sitwy rozmiaru 2, 1 i 2, więc procedura może zwrócić wektor $[2, 1, 2]$.

Przykład 2

Rozważmy następujące wywołanie:

```
partition_players(7, 6, [0, 4, 2, 1, 2, 3], [1, 5, 4, 5, 5, 6])
```

W tym przykładzie przyjaciółmi są gracze 0 i 1, gracze 4 i 5, gracze 2 i 4, gracze 1 i 5, gracze 2 i 5 oraz gracze 3 i 6.

Jedynym przyjacielem gracza 3 jest gracz 6, więc dowolna sitwa złożona z gracza 3 jest albo

- sitwą rozmiaru 1 zawierającą jedynie gracza 3, albo
- sitwą złożoną zarówno z gracza 3 jak i 6.

W tym drugim przypadku sitwa ta musi zawierać także graczy 4 i 5. To nie jest możliwe ponieważ jedynym przyjacielem gracza 6 jest gracz 3, a gracz 3 nie jest połączony z graczami 4 i 5 ciągiem związków przyjaźni.

Gracz 3 musi być więc umieszczony w sitwie rozmiaru 1. Podonie, gracz 6 także musi być umieszony w sitwie rozmiaru 1. Liczba sitw musi zatem wynosić co najmniej 4.

Gracze 0, 1 i 2 nie tworzą sitwy rozmiaru 3, ponieważ ani gracz 0 ani gracz 1 nie są połaczeni z graczem 2 ciągiem związków przyjaźni w ramach sitwy. To nie byłoby prawdą, gdyby gracz 5 był w tej sitwie, ale go tam nie ma. Liczba sitw musi zatem wynosić co najmniej 5.

Z drugiej strony gracze 0 i 1, oraz gracze 4 i 5 tworzą dwie sitwy rozmiaru 2. Kolejkę można zatem podzielić na 5 sitw rozmiaru 2, 1, 1, 2 i 1, więc procedura może zwrócić wektor $[2, 1, 1, 2, 1]$.

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka odczytuje dane wejściowe w następującym formacie:

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

Niech elementami wektora zwróconego przez `partition_players` będą $K[0], K[1], \ldots, K[g-1]$ dla pewnego nieujemnego g. Wyjście przykładowej sprawdzaczki będzie miało następujący format:

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