Загадка Сфінкса

Великий Сфінкс має для вас загадку. Вам дано граф із N вершин. Вершини пронумеровані від 0 до $N-1$. У графі є M ребер, пронумерованих від 0 до $M-1$. Кожне ребро з'єднує пару різних вершин і є двостороннім. Зокрема, для кожного j від 0 до $M-1$ (включно) ребро j з'єднує вершини $X[j]$ і $Y[j]$. Будь-яка пара вершин з'єднана не більше ніж одним ребром. Дві вершини називаються **сусідними** якщо вони з'єднані ребром.

Послідовність вершин $v_0, v_1, \ldots, v_k$ (для $k \geq 0$) називається **шляхом**, якщо кожні дві послідовні вершини v_l і v_{l+1} (для кожного l такого, що $0 \leq l < k$) є сусідами. Ми говоримо, що шлях $v_0, v_1, \ldots, v_k$ **з'єднує** вершини v_0 і v_k. У наданому вам графі кожна пара вершин з'єднана деяким шляхом.

Існує $N+1$ кольорів, пронумерованих від 0 до N. Колір N є особливим і називається **кольором Сфінкса**. Кожній вершині призначається колір. Зокрема, вершина i ($0 \leq i < N$) має колір $C[i]$. Кілька вершин можуть мати однаковий колір, і можуть бути кольори, не призначені жодній вершині. Жодна вершина не має кольору Сфінкса, тобто $0 \leq C[i] < N$ ($0 \leq i < N$).

Шлях $v_0, v_1, \ldots, v_k$ (для $k \geq 0$) називається **однотонним**, якщо всі його вершини мають однаковий колір, тобто $C[v_l] = C[v_{l+1}]$ (для кожного l такого, що $0 \leq l < k$). Крім того, ми говоримо, що вершини p і q ($0 \leq p < N$, $0 \leq q < N$) знаходяться в тій самій **однотонній компоненті** тоді і тільки тоді, коли вони з'єднані однотонним шляхом.

Ви знаєте вершини і ребра, але ви не знаєте, який колір має кожна вершина. Ви хочете дізнатися кольори вершин, виконуючи **експерименти з перефарбуванням**.

В експерименті з перефарбуванням, Ви можете перефарбувати будь-яку кількість вершин. Зокрема, для того, щоб провести експеримент із перефарбуванням ви спочатку маєте вибрати масив E розміром N, де для кожного i ($0 \leq i < N$), $E[i]$ від -1 до N **включно**. Тоді колір кожної вершини i стає $S[i]$, де значення $S[i]$ дорівнює:

- $C[i]$, тобто оригінальний колір i, якщо $E[i] = -1$, або
- $E[i]$, інакше.

Зауважте, що це означає, що ви можете використовувати колір Сфінкса у своєму перефарбуванні.

У кінці Великий Сфінкс оголошує кількість однотонних компонент у графі, після зміни кольору кожної вершини i на $S[i]$ ($0 \leq i < N$). Нове зафарбовування застосовується лише для цього конкретного експерименту з перефарбуванням, тому **кольори всіх вершин повертаються до початкових після завершення експерименту**.

Ваше завдання — визначити кольори вершин у графі, виконуючи не більше $2\,750$ експериментів із перефарбуванням. Ви також можете отримати частковий бал, якщо ви правильно визначите для кожної пари сусідніх вершин, чи мають вони однаковий колір.

Деталі реалізації

Ви повинні реалізувати наступну функцію.

```
std::vector<int> find_colours(int N,
    std::vector<int> X, std::vector<int> Y)
```

- N : кількість вершин у графі.
- X, Y : масиви довжини M, які описують ребра.
- Ця функція має повернути масив G довжини N, що представляє кольори вершин у графі.
- Ця функція викликається рівно один раз для кожного тесту.

Наведена вище функція може викликати наступну функцію для проведення експериментів з перефарбуванням:

```
int perform_experiment(std::vector<int> E)
```

- E : масив довжини N, який визначає, як потрібно змінити кольори вершин.
- Ця функція повертає кількість однотонних компонентів після перефарбування вершин відповідно до E.
- Цю функцію можна викликати не більше $2\,750$ разів.

Градер **не адаптивний**, тобто кольори вершин фіксуються перед викликом `find_colours`.

Обмеження

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $0 \leq X[j] < Y[j] < N$ для кожного j такого, що $0 \leq j < M$.
- $X[j] \neq X[k]$ або $Y[j] \neq Y[k]$ для кожних j і k таких, що $0 \leq j < k < M$.
- Кожна пара вершин з'єднана деяким шляхом.
- $0 \leq C[i] < N$ для кожного i такого, що $0 \leq i < N$.

Підзадачі

Підзадача	Балів	Додаткові обмеження
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Граф є шляхом: $M = N - 1$ і вершини j і $j + 1$ є сусідніми ($0 \leq j < M$).
4	21	Граф повний: $M = \frac{N \cdot (N-1)}{2}$ і будь-які дві вершини є сусідніми.
5	36	Без додаткових обмежень.

У кожній підзадачі ви можете отримати частковий бал, якщо ваша програма визначає правильно для кожної пари сусідніх вершин чи мають вони однаковий колір.

Точніше, ви отримуєте повний бал за підзадачу, якщо у всіх тестових випадках, масив G, який повернула `find_colours`, точно такий же, як масив C (тобто $G[i] = C[i]$, для всіх i таких, що $0 \leq i < N$). інакше, ви отримуєте 50% балу за підзадачу, якщо виконуються такі умови у всіх тестових випадках:

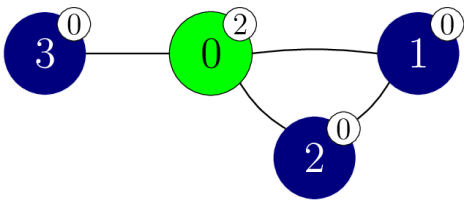
- $0 \leq G[i] < N$ для кожного i такого, що $0 \leq i < N$;
- Для кожного j такого, що $0 \leq j < M$:
  - $G[X[j]] = G[Y[j]]$ тоді і тільки тоді, коли $C[X[j]] = C[Y[j]]$.

Приклад

Розглянемо наступний виклик.

```
find_colours(4, [0, 1, 0, 0], [1, 2, 2, 3])
```

Для цього прикладу припустимо, що (приховані) кольори вершин задані $C = [2, 0, 0, 0]$. Цей сценарій показано на наступному малюнку. Кольори додатково представлені числами на білих мітках, прикріплених до кожної вершини.



Функція може викликати `perform_experiment` наступним чином.

```
perform_experiment([-1, -1, -1, -1])
```

У цьому виклику жодна вершина не змінює колір, оскільки всі вершини зберігають свої початкові кольори.

Розглянемо вершину 1 і вершину 2. Вони обидві мають колір 0, а шлях $1, 2$ є однотонним. У результаті вершини 1 і 2 знаходяться в одній однотонній компоненті.

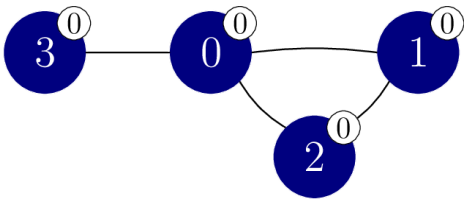
Розглянемо вершину 1 і вершину 3. Хоча обидві мають колір 0, вони знаходяться в різних однотонних компонентах оскільки немає жодного однотонного шляху, що з'єднує їх.

Загалом є 3 однотонних компоненти, з вершинами $\{0\}$, $\{1, 2\}$ і $\{3\}$. Таким чином, цей виклик повертає 3.

Тепер функція може викликати `perform_experiment` наступним чином.

```
perform_experiment([0, -1, -1, -1])
```

У цьому виклику лише вершина 0 змінюється на колір 0, що призводить до зафарбовування, показаного на наступному малюнку.

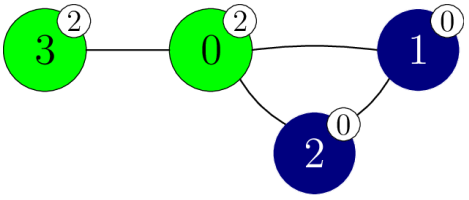


Цей виклик повертає 1, оскільки всі вершини належать одній однотонній компоненті. Тепер ми можемо зробити висновок, що вершини 1, 2 і 3 мають колір 0.

Потім функція може викликати `perform_experiment` наступним чином.

```
perform_experiment([-1, -1, -1, 2])
```

У цьому виклику вершина 3 змінюється на колір 2, що призводить до зафарбування, показаного на наступному малюнку.



Цей виклик повертає 2, оскільки є 2 однотонні компоненти, з вершинами $\{0, 3\}$ і $\{1, 2\}$ відповідно. Ми можемо зробити висновок, що вершина 0 має колір 2.

Потім функція `find_colours` повертає масив $[2, 0, 0, 0]$. Оскільки $C = [2, 0, 0, 0]$, надається повний бал.

Зауважте, що також є кілька повернених значень, для яких буде надано 50% балів, наприклад $[1, 2, 2, 2]$ або $[1, 2, 2, 3]$.

Приклад градера

Формат вхідних даних:

```
N  M
C[0]  C[1] ... C[N-1]
X[0]  Y[0]
X[1]  Y[1]
...
X[M-1]  Y[M-1]
```

Формат вихідних даних:

```
L  Q
G[0]  G[1] ... G[L-1]
```

Тут L — це довжина масиву G, який повертає `find_colours`, а Q — це кількість викликів `perform_experiment`.