Загадката на Сфинкса

Великият Сфинкс има загадка за Вас. Даден Ви е граф с N върха. Върховете са номерирани от 0 до $N-1$. Има M ребра в графа, номерирани от 0 до $M-1$. Всяко ребро свързва два различни върха и е двупосочно. По-конкретно, за всяко j от 0 до $M-1$, ребро j свързва върхове $X[j]$ и $Y[j]$. Между всеки два върха има най-много едно ребро, което ги свързва. Два върха свързани с ребро се наричат **съседни**.

Редица от върхове $v_0, v_1, \ldots, v_k$ (за $k \geq 0$) се нарича **път**, ако всяка двойка поредни върхове v_l и v_{l+1} (за всяко $0 \leq l < k$) са съседни. Казваме, че пътя $v_0, v_1, \ldots, v_k$ **свързва** върхове v_0 и v_k. В дадения Ви граф всеки два върха са свързани с някой път.

Има $N+1$ цвята номерирани от 0 до N. Цвят N е специален и се нарича **цвета на Сфинкса**. Всеки връх има цвят. По-конкретно, връх i ($0 \leq i < N$) има цвят $C[i]$. Множество върхове може да имат един и същи цвят. Също така може да има цветове, за които няма върхове оцветени в тези цветове. Никой връх не е оцветен в цвета на Сфинкса, т.е. $0 \leq C[i] < N$ ($0 \leq i < N$).

Пътят $v_0, v_1, \ldots, v_k$ (за $k \geq 0$) се нарича **едноцветен**, ако всички върхове в него са в един и същи цвят, т.е. $C[v_l] = C[v_{l+1}]$ (за всяко $0 \leq l < k$). Също така, казваме, че върхове p и q ($0 \leq p < N$, $0 \leq q < N$) са в една и съща **едноцветна компонента**, тогава и само тогава когато са свързани с едноцветен път.

Вие знаете върховете и ребрата, но не знаете цветовете на върховете. Искате да откриете цветовете на върховете като извършвате **експерименти с преоцветяване**.

В един експеримент с преоцветяване Вие можете да смените цвета на произволно много върхове. По-конкретно, за да извършите експеримент с преоцветяване, първо избирате редица E с дължина N, където за всяко $0 \leq i < N$, $E[i]$ е между -1 и N **включително**. Тогава, цветът на връх i става $S[i]$, където $S[i]$ е равно на:

- $C[i]$, т.е. оригиналния цвят на връх i, ако $E[i] = -1$, или
- $E[i]$, иначе.

Забележете, че това означава, че можете да използвате цвета на Сфинкса във Вашето преоцветяване.

Великият Сфинкс обявява броя монохроматични компоненти в графа, след като смени цвета на всеки връх i на $S[i]$ ($0 \leq i < N$). Новото преоцветяване се прилага само за този експеримент с преоцветяване, т.е. **цветовете на всички върхове се връщат на оригиналните им такива след края на експеримента**.

Вашата задача е да откриете цветовете на върховете в графа като извършите най-много $2\,750$ експеримента с преоцветяване. Също така, ще получите частични точки, ако успешно откриете, за всяка двойка съседни върхове, дали са в един и същи цвят.

Детайли по имплементацията

Трябва да напишете следната функция:

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

- N: броят върхове в графа.
- X, Y: вектори с дължина M, описващи ребрата.
- Тази функция трябва да върне вектор G с дължина N, който описва цветовете на върховете в графа.
- Тя се вика точно веднъж.

Горната функция може да извиква следната функция, за да прави експерименти с преоцветяване:

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

- E: вектор с дължина N, описващ как върховете да бъдат преоцветени.
- Тази функция връща броя на монохроматичните компоненти след преоцветяване на върховете според E.
- Тя може да бъде извикана най-много $2\,750$ пъти.

Грейдърът **не е адаптивен**, т.е. цветовете на върховете са избрани преди извикването на `find_colours`.

Ограничения

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $0 \leq X[j] < Y[j] < N$ за всяко $0 \leq j < M$.
- $X[j] \neq X[k]$ или $Y[j] \neq Y[k]$ за всички $0 \leq j < k < M$.
- Всяка двойка върхове е свързана с някой път.
- $0 \leq C[i] < N$ за всяко $0 \leq i < N$.

Подзадачи

Подзадача	Точки	Допълнителни ограничения
1	3	$N = 2$
2	7	$N \le 50$
3	33	Графът е пътека: $M = N - 1$ и върхове j и $j+1$ са съседни ($0 \le j < M$).
4	21	Графът е пълен: $M = \frac{N \cdot (N-1)}{2}$ и всеки два върха са съседни.
5	36	Няма.

Във всяка подзадача, ще получите частични точки, ако Вашата програма правилно открие, за всяка двойка свързани върхове, дали имат един и същи цвят.

По-конкретно, получавате пълните точки за дадена подзадача, ако във всички тестове, векторът G върнат от `find_colours` е точно същият като вектора C (т.е. $G[i] = C[i]$ за всяко $0 \le i < N$). Иначе, получавате 50% от точките за подзадачата, ако следните условия важат за всички тестове:

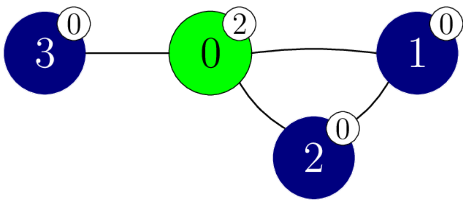
- $0 \le G[i] < N$ за всяко $0 \le i < N$;
- За всяко $0 \le j < M$:
    - $G[X[j]] = G[Y[j]]$, тогава и само тогава когато $C[X[j]] = C[Y[j]]$.

Пример

Нека разгледаме следното извикване:

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

В този пример, да приемем, че (скритите) цветове на върховете са: $C = [2, 0, 0, 0]$. Този случай е показан по-долу. Цветовете са допълнително представени с числа в белите кръгчете до всеки връх.



Функцията може да извика `perform_experiment` по следния начин:

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

В това извикване, никой връх не е преоцветен и всички остават в оригиналния си цвят.

Да разгледаме върхове 1 и 2. И двата имат цвят 0 и пътя $1, 2$ е едноцветен. Следва, че 1 и 2 са в една и съща едноцветна компоеннта.

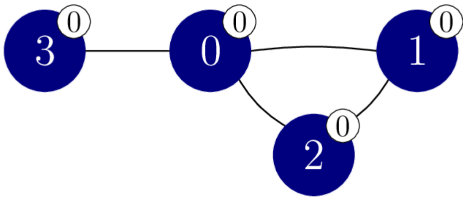
Да разгледаме върхове 1 и 3. Въпреки че и двата са от цвят 0, те са в различни едноцветни компоненти, защото няма едноцветен път, който да ги свързва.

Общо има 3 едноцветни компоентни, с върхове: $\{0\}$, $\{1, 2\}$ и $\{3\}$. Следва, че това извикване ще върне 3.

След това, може да се извика `perform_experiment` така:

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

В това извикване, само връх 0 е преоцветен към цвят 0, който води до следното оцветяването:

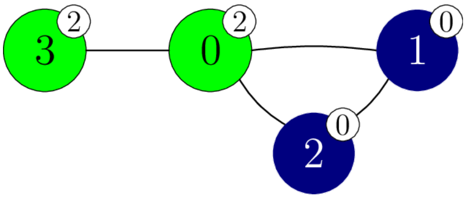


Това извикване връща 1, защото всички върхове са в една и съща едноцветна компонента. Следва, че можем да заключим, че върхове 1, 2, и 3 имат цвят 0.

После може да бъде извикано `perform_experiment` по следния начин:

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

В това извикване връх 3 е преоцветен към цвят 2, което води до следното оцветяване:



Това извикване връща 2, защото има 2 едноцветни компоненти, с върхове $\{0, 3\}$ и $\{1, 2\}$. Можем да дедуцираме, че връх 0 има цвят 2.

Най-накрая, функцията `find_colours` връща вектора $[2, 0, 0, 0]$. Тъй като това съвпада с $C = [2, 0, 0, 0]$, се получават пълните точки.

Забележете, че има много възможни стойности на върнатия вектор, за които биха се получили 50% от точките, например: $[1, 2, 2, 2]$ или $[1, 2, 2, 3]$.

Локален грейдър

Входен формат:

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

Изходен формат:

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

Тук L е дължината на вектора G върнат от `find_colours`, а Q е броят извиквания на `perform_experiment`.