

## Загатката на Сфингата

Големата Сфинга има загатка за вас. Даден е граф со  $N$  темиња. Темињата се нумерирани со целите броеви од 0 до  $N - 1$ . Има  $M$  ребра во графот, нумерирани со целите броеви од 0 до  $M - 1$ . Секое ребро поврзува пар различни темиња и е двонасочно. Поточно, за секое  $j$  од 0 до  $M - 1$  (вклучително), реброто  $j$  ги поврзува темињата  $X[j]$  и  $Y[j]$ . Постои најмногу едно ребро што поврзува кој било пар темиња. Две темиња се нарекуваат **соседни** ако се поврзани со ребро.

Низа од темиња  $v_0, v_1, \dots, v_k$  (за  $k \geq 0$ ) се нарекува **пат** ако секои две последователни темиња  $v_l$  и  $v_{l+1}$  (за секое  $l$  такво што  $0 \leq l < k$ ) се соседни. Велиме дека патот  $v_0, v_1, \dots, v_k$  **ги поврзува** темињата  $v_0$  и  $v_k$ . Во графот што ви е даден, секој пар темиња е поврзан со некој пат.

Постојат  $N + 1$  бои, нумерирани со целите броеви од 0 до  $N$ . Бојата  $N$  е специјална и се нарекува **Боја на Сфингата**. На секое теме му се доделува боја. Конкретно, темето  $i$  ( $0 \leq i < N$ ) има боја  $C[i]$ . Повеќе темиња може да имаат иста боја, и може да има бои кои не се доделени на ниту едно теме. Ниту едно теме ја нема бојата на Сфингата, односно  $0 \leq C[i] < N$  ( $0 \leq i < N$ ).

Патот  $v_0, v_1, \dots, v_k$  (за  $k \geq 0$ ) се нарекува **монохроматски** ако сите негови темиња имаат иста боја, т.е.  $C[v_l] = C[v_{l+1}]$  (за секое  $l$  такво што  $0 \leq l < k$ ). Дополнително, велиме дека темињата  $p$  и  $q$  ( $0 \leq p < N$ ,  $0 \leq q < N$ ) се во иста **монохроматска компонента** ако и само ако се поврзани со монохроматски пат.

Ги знаете темињата и ребрата, но не знаете која боја ја има секое теме. Сакате да ги дознаете боите на темињата, со изведување **експерименти на пребојување**.

Во даден експеримент на пребојување, може да пребоите произволно многу темиња. Конкретно, за да изведете експеримент на пребојување, прво избирате низа  $E$  со големина  $N$ , каде за секое  $i$  ( $0 \leq i < N$ ),  $E[i]$  е помеѓу  $-1$  и  $N$  **вклучително**. Потоа, бојата на секое теме  $i$  станува  $S[i]$ , каде што вредноста на  $S[i]$  е:

- $C[i]$ , односно оригиналната боја на  $i$ , ако  $E[i] = -1$ , или
- $E[i]$ , во спротивно.

Да забележиме дека ова значи дека можете да ја користите бојата на Сфингата во вашето пребојување.

Конечно, Големата Сфинга го објавува бројот на монохроматски компоненти во графот, откако ќе се постави бојата на секое теме  $i$  на  $S[i]$  ( $0 \leq i < N$ ). Новото бојење се применува само за овој конкретен експеримент на пребојување, па **боите на сите темиња се враќаат на првобитните откако ќе заврши експериментот**.

Ваша задача е да ги идентификувате боите на темињата во графот со изведување на најмногу 2 750 експерименти на пребојување. Може, исто така, да добиете и делумни поени, ако за секој пар соседни темиња одредите точно дали ја имаат истата боја.

## Имплементациски детали

Треба да ја имплементирате следната процедура.

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

- $N$ : бројот на темиња во графот.
- $X, Y$ : низи со должина  $M$  што ги опишуваат ребрата.
- Оваа процедура треба да врати низа  $G$  со должина  $N$ , што ги претставува боите на темињата во графот.
- Оваа процедура се повикува точно еднаш за секој тест случај.

Горната процедура може да прави повици до следната процедура за да изведува експерименти на пребојување:

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

- $E$ : низа со должина  $N$  која специфицира како треба да се пребојат темињата.
- Оваа процедура го враќа бројот на монохроматски компоненти после пребојување на темињата според низата  $E$ .
- Оваа процедура може да се повика најмногу 2 750 пати.

Оценувачот **не е адаптивен**, т.е. боите на темињата се фиксирани пред да се направи повик до `find_colours`.

## Ограничувања

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $0 \leq X[j] < Y[j] < N$  за секое  $j$  такво што  $0 \leq j < M$ .
- $X[j] \neq X[k]$  или  $Y[j] \neq Y[k]$  за секои  $j$  и  $k$  такви што  $0 \leq j < k < M$ .
- Секој пар од темиња е поврзан со некој пат.
- $0 \leq C[i] < N$  за секое  $i$  такво што  $0 \leq i < N$ .

## Подзадачи

Подзадача	Поени	Дополнителни ограничувања
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Графот е пат: $M = N - 1$ и темињата $j$ и $j + 1$ се соседни ( $0 \leq j < M$ ).
4	21	Графот е комплетен: $M = \frac{N \cdot (N-1)}{2}$ и кои било две темиња се соседни (поврзани со ребро).
5	36	Нема дополнителни ограничувања.

Во секоја подзадача, може да освоите делумни поени ако вашата програма за секој пар од соседни темиња определува точно дали тие ја имаат истата боја.

Попрецизно, ги добивате сите поени за дадена подзадача ако во сите нејзини тест случаи, низата  $G$  вратена од `find_colours` е точно иста со низата  $C$  (т.е.  $G[i] = C[i]$  за секое  $i$  така што  $0 \leq i < N$ ). Инаку, добивате 50% од поените за дадена подзадача ако се исполнети следните услови во сите нејзини тест случаи:

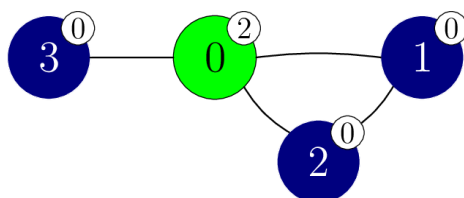
- $0 \leq G[i] < N$  за секое  $i$  такво што  $0 \leq i < N$ ;
- За секое  $j$  такво што  $0 \leq j < M$ :
  - $G[X[j]] = G[Y[j]]$  ако и само ако  $C[X[j]] = C[Y[j]]$ .

## Пример

Да го разгледаме следниот повик.

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

За овој пример, да претпоставиме дека (скриените) боите на темињата се дадени со  $C = [2, 0, 0, 0]$ . Ова сценарио е прикажано на следната слика. Боите дополнително се претставени со броеви во бели кругчиња прикачени на секое теме.



Процедурата може да ја повика `perform_experiment` како што следува.

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

Во овој повик, ниту едно темето не е пребоено, бидејќи сите темиња ја задржуваат својата оригинална боја.

Да ги разгледаме темето 1 и темето 2. Двете ја имаат бојата 0 и патот 1,2 е монохроматски пат. Како резултат на тоа, темињата 1 и 2 се во истата монохроматска компонента.

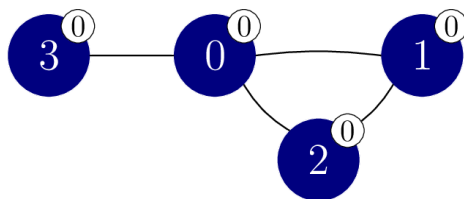
Да ги разгледаме темето 1 и темето 3. Иако и двете темиња ја имаат бојата 0, тие се во различни монохроматски компоненти бидејќи не постои монохроматски пат што ги поврзува нив.

Севкупно, постојат 3 монохроматски компоненти, со темиња  $\{0\}$ ,  $\{1,2\}$  и  $\{3\}$ . Според тоа, овој повик враќа 3.

Сега процедурата може да ја повика `perform_experiment` како што следува.

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

Во овој повик, само темето 0 се пребојува во бојата 0, што резултира во обојувањето прикажано на следната слика.

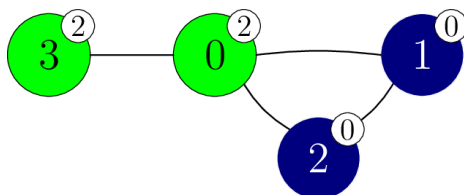


Овој повик враќа 1, бидејќи сите темиња припаѓаат на истата монохроматска компонента. Сега можеме да заклучиме дека темињата 1, 2 и 3 ја имаат бојата 0.

Процедурата потоа може да ја повика `perform_experiment` како што следува.

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

Во овој повик, темето 3 се пребојува во бојата 2, што резултира во обојувањето прикажано на следната слика.



Овој повик враќа 2, бидејќи постојат 2 монохроматски компоненти, со темиња  $\{0, 3\}$  и  $\{1, 2\}$ , соодветно. Можеме да заклучиме дека темето 0 ја има бојата 2.

Процедурата `find_colours` потоа ја враќа низата  $[2, 0, 0, 0]$ . Бидејќи  $C = [2, 0, 0, 0]$ , се доделуваат сите поени.

Да забележиме дека исто така постојат и повеќе повратни вредности, за кои би се доделиле 50% од поените, на пример  $[1, 2, 2, 2]$  или  $[1, 2, 2, 3]$ .

## Пример-оценувач

Формат на влез:

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

Формат на излез:

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

Овде,  $L$  е должината на низата  $G$  вратена од `find_colours`, а  $Q$  е бројот на повици до `perform_experiment`.