Sfinksin Tapmacası

Böyük Sfinksin sizin üçün bir tapmacası var. Sizə N təpəli qraf verilir. Təpələr 0 ilə $N-1$ arasında nömrələnib. Qrafda 0 ilə $M-1$ arasında nömrələnmiş M bağlantı var. Hər bir bağlantı iki fərqli təpəni birləşdirir və iki istiqamətlidir. Konkret olaraq, 0 və $M-1$ (daxil olmaqla) aralığında olan hər j üçün j bağlantısı $X[j]$ və $Y[j]$ təpələrini birləşdirir. Hər hansı iki təpəni birləşdirən ən çox bir bağlantı var. İki təpə bağlantı ilə birləşdirilirsə, onlar **qonşu** adlanır.

$v_0, v_1, \ldots, v_k$ təpələri ardıcıllığındakı ($k \geq 0$ üçün) bütün iki ardıcıl v_l və v_{l+1} təpələri ($0 \leq l < k$ ödəyən bütün l dəyərləri üçün) qonşu olarsa bu ardıcıllıq **yol** adlanır. Biz deyirik ki, $v_0, v_1, \ldots, v_k$ yolu v_0 və v_k təpələrini **birləşdirir**. Sizə verilən qrafda istənilən iki təpə hansısa yol ilə birləşir.

0-dan N-ə qədər nömrələnmiş $N+1$ rəng var. N rəngi xüsusidir və **Sfinksin rəngi** adlanır. Hər bir təpəyə bir rəng verilir. Konkret olaraq, i təpəsi ($0 \leq i < N$) $C[i]$ rənginə malikdir. Birdən çox təpə eyni rəngə malik ola bilər və bəzi rənglər heç bir təpəyə təyin olunmaya bilər. Heç bir təpədə Sfenksin rəngi yoxdur, yəni $0 \leq C[i] < N$ ($0 \leq i < N$).

$v_0, v_1, \ldots, v_k$ ($k \geq 0$ üçün) yolundakı bütün təpələr eyni rəngdədirsə, yəni $C[v_l] = C[v_{l+1}]$ ($0 \leq l < k$ ödəyən hər l üçün) olarsa bu yol **monoxromatik** adlanır. Əlavə olaraq, p və q təpələri ($0 \leq p < N$, $0 \leq q < N$) yalnız və yalnız monoxromatik yol ilə birləşərsə onlar bir **monoxromatik komponentdə** yerləşir.

Siz təpələri və bağlantıları bilirsiniz, lakin hər təpənin hansı rəngə malik olduğunu bilmirsiniz. Siz **yenidən rəngləmə təcrübələri** həyata keçirərək təpələrin rənglərini öyrənmək istəyirsiniz.

Yenidən rəngləmə təcrübəsində siz ixtiyari olaraq bir çox təpələri yenidən rəngləyə bilərsiniz. Xüsusilə, yenidən rəngləmə təcrübəsini yerinə yetirmək üçün əvvəlcə N ölçüsündə E massivi seçməlisiniz, burada hər i ($0 \leq i < N$) üçün $E[i]$ -1 və N (**daxil olmaqla**) aralığındadır. Sonra hər i təpəsinin rəngi $S[i]$ olur, burada $S[i]$ dəyəri:

- əgər $E[i] = -1$ olarsa, $C[i]$, yəni i-nin orijinal rəngi, və ya
- əks halda, $E[i]$ olur.

Nəzərə alın ki, siz yenidən rəngləmənizdə Sfinksin rəngindən istifadə edə bilərsiniz.

Nəhayət, Böyük Sfinks hər bir i təpəsinin rəngini ($0 \leq i < N$) $S[i]$ etdikdən sonra qrafdakı monoxromatik komponentlərin sayını elan edir. Yeni rəngləmə yalnız bu xüsusi yenidən rəngləmə təcrübəsi üçün tətbiq edilir, beləliklə **təcrübə bitdikdən sonra bütün təpələrin rəngləri orijinallarına qayıdır**.

Tapşırığınız ən çox $2\,750$ yenidən rəngləmə təcrübəsi həyata keçirərək qrafdakı təpələrin rənglərini müəyyən etməkdir. Hər bir qonşu təpə cütü üçün onların eyni rəngə malik olub-olmadığını düzgün müəyyən etsəniz, siz həm də qismən bal ala bilərsiniz.

İcra Təfərrüatları

Aşağıdakı proseduru yerinə yetirməlisiniz.

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

- N: qrafdakı təpələrin sayı.
- X, Y: bağlantıları göstərən M uzunluqlu massivlər.
- Bu prosedur geriyə qrafdakı təpələrin rənglərini gostərən N uzunluqlu G massivini qaytarmalıdır.
- Bu prosedur hər bir test üçün bir dəfə çağrılır.

Bu prosedur yenidən rəngləmə təcrübəsi həyata keçirmək üçün aşağıdakı proseduru çağıra bilər:

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

- E: Təpələrin necə yenidən rəngləndiyini göstərən N uzunluqlu massiv.
- Bu prosedur E-yə uyğun olaraq təpələri yenidən rənglədikdən sonra monoxromatik komponentlərin sayını geri qaytarır.
- Bu prosedur ən çox $2\,750$ dəfə çağrıla bilər.

Qiymətləndirici **adaptiv deyil**, yəni təpələrin rəngləri `find_colours` çağırışından əvvəl müəyyən edilir.

Məhdudiyyətlər

- $2 \le N \le 250$
- $N - 1 \le M \le \frac{N \cdot (N-1)}{2}$
- Hər bir $0 \le j < M$ bərabərsizliyinə uyğun j üçün $0 \le X[j] < Y[j] < N$.
- Hər bir $0 \le j < k < M$ bərabərsizliyinə uyğun j və k üçün $X[j] \neq X[k]$ və ya $Y[j] \neq Y[k]$.
- Hər bir təpə cütü yol ilə birləşir.
- Hər bir $0 \le i < N$ bərabərsizliyinə uyğun i üçün $0 \le C[i] < N$.

Alt Tapşırıqlar

Alt Tapşırıq	Bal	Əlavə Məhdudiyyətlər
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Qraf bir yoldur: $M = N - 1$, və j və $j + 1$ təpələri qonşudurlar ($0 \leq j < M$).
4	21	Qraf tam qrafdır: $M = \frac{N \cdot (N-1)}{2}$ və istənilən iki təpə qonşudurlar.
5	36	Əlavə məhdudiyyət yoxdur.

Proqramınız hər bir qonşu təpə cütü üçün onların eyni rəngə malik olub-olmadığını düzgün müəyyən edərsə, hər bir alt tapşırıqda siz qismən xal əldə edə bilərsiniz.

Daha dəqiq desək, əgər bir alt tapşırıqdakı bütün test vəziyyətlərində `find_colours` tərəfindən qaytarılan G massivi C massivi ilə tam eyni olarsa, siz alt tapşırığın bütün xalını əldə etmiş olursunuz (yəni $0 \leq i < N$ ödəyən hər bir i üçün $G[i] = C[i]$). Əks halda, aşağıdakı şərtlər bütün testlərdə doğru olarsa, alt tapşırıq üçün 50% bal alırsınız:

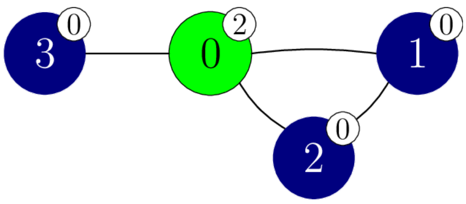
- Hər bir $0 \leq i < N$ bərabərsizliyinə uyğun i üçün $0 \leq G[i] < N$;
- Hər bir $0 \leq j < M$ bərabərsizliyinə uyğun j üçün:
  - $G[X[j]] = G[Y[j]]$ yalnız və yalnız $C[X[j]] = C[Y[j]]$ olduqda.

Nümunə

Aşağıdakı çağırışı nəzərdən keçirin.

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

Bu misal üçün fərz edək ki, təpələrin (gizli) rəngləri $C = [2, 0, 0, 0]$ ilə verilir. Bu ssenari aşağıdakı şəkildə göstərilmişdir. Rənglər əlavə olaraq hər bir təpəyə əlavə edilmiş ağ etiketlərdə rəqəmlərlə təmsil olunur.



Prosedur aşağıdakı kimi `perform_experiment` çağıra bilər.

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

Bu çağırışda heç bir təpə yenidən rənglənmir, bütün təpələr orijinal rənglərini saxlayır.

1 və 2 təpəsini nəzərdən keçirin. Hər ikisi 0 rənginə malikdir və $1, 2$ yolu monoxromatik yoldur. Nəticədə, 1 və 2 təpələri eyni monoxromatik komponentdədir.

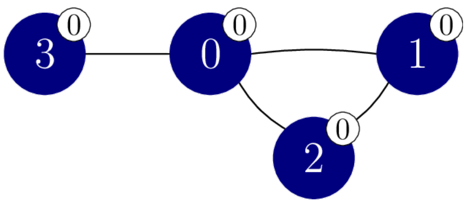
1 və 3 təpəsini nəzərdən keçirin. Onların hər ikisinin rəngi 0 olsa da, onları birləşdirən monoxromatik yol olmadığı üçün onlar fərqli monoxromatik komponentlərdədir.

Ümumilikdə təpələri $\{0\}$, $\{1, 2\}$ və $\{3\}$ olan 3 monoxromatik komponent var. Beləliklə, bu çağırış 3 qaytarır.

İndi prosedur `perform_experiment` çağırışını belə edə bilər.

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

Bu çağırışda yalnız 0 təpəsi 0 rənginə dəyişdirilir ki, bu da aşağıdakı şəkildə göstərilən rəngləmə ilə nəticələnir.

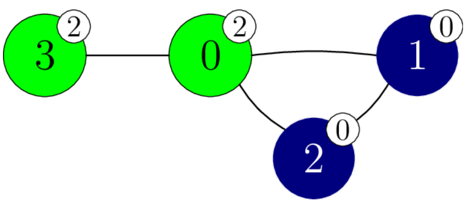


Bu çağırış 1 qaytarır, çünki bütün təpələr eyni monoxromatik komponentə aiddir. İndi belə nəticə çıxara bilərik ki, 1 , 2 və 3 təpələri 0 rənginə malikdirlər.

Sonra prosedur `perform_experiment` çağırışını belə edə bilər.

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

Bu çağırışda 3 təpəsi 2 rənginə yenidən rənglənir ki, bu da aşağıdakı şəkildə göstərilən rəngləmə ilə nəticələnir.



Bu çağırış 2 qaytarır, çünki ucları müvafiq olaraq $\{0, 3\}$ və $\{1, 2\}$ olan 2 monoxromatik komponent var. Biz belə nəticə çıxara bilərik ki, 0 təpəsi 2 rənginə malikdir.

Daha sonra `find_colours` proseduru $[2, 0, 0, 0]$ massivini geri qaytarır. $C = [2, 0, 0, 0]$ olduğundan tam bal verilir.

Nəzərə alın ki, balın 50%-i veriləcək bir neçə geri qaytarıla bilən dəyərlər də var (məsələn $[1, 2, 2, 2]$ və ya $[1, 2, 2, 3]$).

Nümunə Qiymətləndirici

Giriş formatı:

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

Çıxış formatı:

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

Burada, L `find_colours` tərəfindən qaytarılan G massivinin uzunluğu, Q isə `perform_experiment` üçün edilən çağırışların sayıdır.