Szfinx rejtvénye

A Nagy Szfinxnek van egy rejtvénye számodra. Kapsz egy gráfot, aminek N csúcsa és M éle van. A csúcsok számozása 0 és $N-1$ közötti, az éleké 0 és $M-1$ közötti. A j. él az $X[j]$ és az $Y[j]$ csúcsokat köti össze. Mindegyik él két különböző csúcsot köt össze és kétirányú.

Két csúcsot **szomszédosnak** nevezünk, ha éllel vannak összekötve. Bármely csúcspárt legfeljebb egy él köt össze.

Csúcsoknak egy $v_0, v_1, \ldots, v_k$ ($k \geq 0$) sorozatát **útnak** hívjuk, ha benne minden egymást követő két csúcs, v_l és v_{l+1} ($0 \leq l < k$) szomszédos. Azt mondjuk, hogy a $v_0, v_1, \ldots, v_k$ út **összeköti** a v_0 és a v_k csúcsokat. A megadott gráfban minden csúcspárt összeköt valamely út.

Van $N+1$ szín, 0 és N közötti számokkal jelölve. Az N értékű szín különleges, a **Szfinx színének** hívják. Minden csúcshoz hozzá van rendelve egy szín. Pontosabban, az i ($0 \leq i < N$) csúcs színe $C[i]$. Több csúcs is lehet azonos színű, és lehetnek olyan színek, amelyek nincsenek hozzárendelve egyetlen csúcshoz sem. Egyetlen csúcsnak a színe sem a Szfinx színe, vagyis $0 \leq C[i] < N$ ($0 \leq i < N$).

A $v_0, v_1, \ldots, v_k$ utat ($k \geq 0$) **monokromatikusnak** hívjuk, ha minden csúcsa azonos színű, azaz $C[v_l] = C[v_{l+1}]$ minden l-re, ahol $0 \leq l < k$. Ezenkívül azt mondjuk, hogy a p és a q csúcsok ($0 \leq p < N$, $0 \leq q < N$) akkor és csak akkor vannak ugyanabban az **monokromatikus komponens**ben, ha monokromatikus út köti össze őket.

Ismered a csúcsokat és az éleket, de nem ismered a csúcsok színeit. Meg kell határoznod a csúcsok színeit **átszínezési kísérlet**ek elvégzésével.

Egy átszínezési kísérletben tetszőlegesen átszínezhetsz akárhány csúcsot. Pontosabban egy átszínezési kísérlethez meg kell adni egy N elemű E sorozatot, ahol $-1 \leq E[i] \leq N$ minden i-re ($0 \leq i < N$). Ezzel a sorozattal elvégezve az átszínezést, minden i csúcs színe $S[i]$ lesz, ahol a $S[i]$ értéke:

- $C[i]$, azaz az i eredeti színe, ha $E[i] = -1$,
- $E[i]$, egyébként.

Vedd figyelembe, hogy ez azt jelenti, hogy az átszínezésnél használhatod a Szfinx színét is.

Az átszínezést után a Nagy Szfinx bejelenti a monokromatikus komponensek számát.

Az új színezés csak erre a bizonyos átszínezési kísérletre vonatkozik, így **a kísérlet befejezése után az összes csúcs színe visszaáll az eredetire**.

Az a feladatod, hogy meghatározd a gráf csúcsainak színeit legfeljebb $2\,750$ átszínezési kísérlet elvégzésével. Részpontszámot kaphatsz, ha helyesen határozod meg minden szomszédos csúcspárra, hogy azonos színűek-e.

Megvalósítás

A következő eljárást kell megvalósítanod.

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

- N : a gráf csúcsainak száma.
- X , Y: az éleket megadó M elemű tömbök.
- Ennek az eljárásnak egy N hosszúságú G tömböt kell visszaadnia, ami megadja a gráf csúcsainak a színét.
- Ezt az eljárást minden tesztesetre pontosan egyszer hívják meg.

A fenti eljárás a következő eljárást hívhatja az átszínezési kísérletek elvégzéséhez:

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

- E : N hosszúságú tömb, amely megadja, hogy a csúcsokat hogyan kell átszínezni.
- Ez az eljárás a csúcsok E szerinti átszínezése utáni monokromatikus komponensek számát adja vissza.
- Ez az eljárás legfeljebb $2\,750$ alkalommal hívható meg.

Az értékelő **nem adaptív**, azaz a csúcsok színei rögzítve vannak a `find_colours` hívása előtt.

Korlátok

- $2 \le N \le 250$
- $N - 1 \le M \le \frac{N \cdot (N-1)}{2}$
- $0 \le X[j] < Y[j] < N$ minden j-re, ahol $0 \le j < M$.
- $X[j] \ne X[k]$ vagy $Y[j] \ne Y[k]$ minden j és k esetén, ahol $0 \le j < k < M$.
- Minden csúcspárt valamilyen útvonal köt össze.
- $0 \le C[i] < N$ minden i-re, ahol $0 \le i < N$.

Részfeladatok

Részfeladat	Pontszám	További megszorítások
1	3	$N = 2$
2	7	$N \leq 50$
3	33	A gráf egy útvonal: $M = N - 1$ és a j és a $j+1$ csúcsok szomszédosak ($0 \leq j < M$)
4	21	A gráf teljes: $M = \frac{N \cdot (N-1)}{2}$ és bármely két csúcs szomszédos
5	36	Nincsenek további megszorítások

Minden részfeladatban részpontszámot kaphatsz, ha a programod helyesen határozza meg minden szomszédos csúcspárra, hogy azonos színűek-e. Pontosabban, megkapod egy részfeladat teljes pontszámát, ha minden tesztesetben, `find_colours` által visszaadott G tömb pontosan ugyanaz, mint a C tömb (azaz $G[i] = C[i]$ minden i-re, ahol $0 \leq i < N$). Egyébként, egy részfeladat pontszámából 50%-ot kapsz, ha a következő feltételek állnak fenn minden tesztesetben:

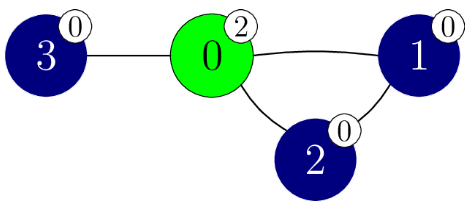
- $0 \leq G[i] < N$ minden i-re, ahol $0 \leq i < N$;
- Minden j-re, ahol $0 \leq j < M$:
  - $G[X[j]] = G[Y[j]]$ akkor és csak akkor, ha $C[X[j]] = C[Y[j]]$.

Példa

Tekintsük a következő hívást.

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

Ebben a példában tegyük fel, hogy a csúcsok (rejtett) színei: $C = [2, 0, 0, 0]$. Ez az eset a következő ábrán látható. A színek értékét az egyes csúcsokhoz csatolt fehér címkéken látjuk.



Az eljárás a következőképpen hívhatja meg `perform_experiment` .

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

Ebben a hívásban egyetlen csúcs sem színeződik át, mivel minden csúcs megtartja eredeti színét.

Tekintsük az 1 és a 2 csúcsot. Mindkettő színe 0 és az $1, 2$ útvonal egy monokromatikus útvonal. Ennek eredményeként 1 és 2 csúcsok ugyanabban a monokromatikus komponensben vannak.

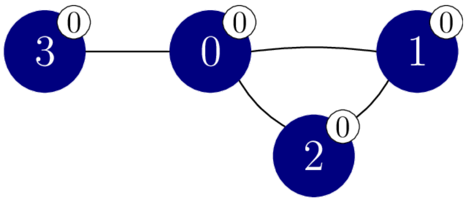
Tekintsük az 1 és a 3 csúcsot. Bár mindkettőnek 0 a színe, különböző monokromatikus komponensekben vannak, mivel nem köti össze őket monokromatikus út.

Összességében 3 monokromatikus komponens van, $\{0\}$, $\{1, 2\}$ és $\{3\}$ csúcsokkal. Így ez a hívás 3 -at ad vissza.

Az eljárás a következőképpen hívhatja meg a `perform_experiment` eljárást.

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

Ebben a hívásban csak a 0 csúcs van átszínezve 0 színre, ami a következő ábrán látható színezést eredményezi.

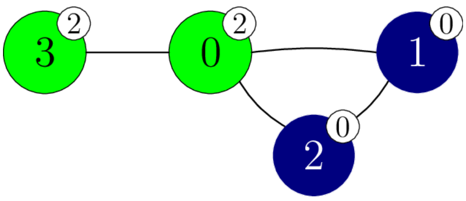


Ez a hívás 1 értékkel tér vissza, mivel az összes csúcs ugyanahhoz a monokromatikus komponenshez tartozik. Most már arra következtethetünk, hogy az 1 , a 2 és a 3 csúcsok színe 0 .

Az eljárás ezután meghívhatja `perform_experiment`-et a következőképpen.

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

Ebben a hívásban a 3 csúcs átszíneződik 2 színre, ami a következő ábrán látható színezést eredményezi.



Ez a hívás 2 értéket ad vissza, mivel 2 monokromatikus komponens van, $\{0, 3\}$ és $\{1, 2\}$ csúcsokkal. Ebből arra következtethetünk, hogy a 0 csúcs színe 2 .

A `find_colours` eljárás ezután a $[2, 0, 0, 0]$ tömböt adja vissza. Mivel $C = [2, 0, 0, 0]$, a teljes pontszámot kapod.

Vedd figyelembe, hogy több visszatérési érték is lehet, amelyekre a pontszám 50%-át kapnád, például $[1, 2, 2, 2]$ vagy $[1, 2, 2, 3]$.

Mintaértékelő

Beviteli formátum:

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

Kimeneti formátum:

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

Itt L a `find_colours` által visszaadott G tömb hossza, és Q a `perform_experiment` hívások száma.