

Hádanka bratislavského koňa

Pred bratislavským hradom stojí socha koňa (a na ňom Svätopluk, ten je ale pre našu úlohu nepodstatný). Bratislavský kôň bol na návšteve Egypta, kde sa mu veľmi zapáčila sfinga. Po návrate domov preto pripravil pre návštevníkov hradu hádanku. Keďže bol ovplyvnený hordou informatikov na egyptských plážach, je jeho hádanka pre obyčajných ľudí príliš ťažká. Musíte im preto vy pomôcť dostať sa do hradu.

Kôň vám zadá graf s N vrcholmi, ktoré sú očíslované od 0 po $N - 1$. V grafe sa nachádza M neorientovaných hrán očíslovaných od 0 po $M - 1$. Každá hrana spája dva rôzne vrcholy. Konkrétne, pre všetky j od 0 po $M - 1$ spája hrana j vrcholy $X[j]$ a $Y[j]$. V koňovom grafe je každá dvojica vrcholov priamo spojená najviac jednou hranou. Dva vrcholy voláme **susedné**, ak sú priamo spojené hranou.

Postupnosť navzájom rôznych vrcholov v_0, v_1, \dots, v_k (pre $k \geq 0$) nazveme **cestou** ak platí, že pre každé l také, že $0 \leq l < k$, sú vrcholy v_l a v_{l+1} susedné. V zadanom grafe platí, že medzi ľubovoľnými dvoma vrcholmi existuje aspoň jedna cesta, ktorá ich spája.

Kôň používa $N + 1$ rôznych farieb očíslovaných od 0 po N . Farba N je špeciálna **konská farba**. Každý vrchol grafu má priradenú jednu farbu. Konkrétne, vrchol i (pre $0 \leq i < N$) má farbu $C[i]$. Ofarbenie môže byť ľubovoľné: viacero vrcholov môže mať priradenú rovnakú farbu a niektoré farby môžu ostať nepoužité. Navyše viete, že **žiadne** vrchol nemá konskú farbu, teda $0 \leq C[i] < N$ pre $0 \leq i < N$.

Cesta v_0, v_1, \dots, v_k (pre $k \geq 0$) je **jednofarebná** ak majú všetky vrcholy tejto cesty rovnakú farbu, t.j. $C[v_l] = C[v_{l+1}]$ (pre každé l také, že $0 \leq l < k$). O dvoch vrchoch p a q ($0 \leq p < N$, $0 \leq q < N$) hovoríme, že sú v spoločnom **jednofarebnom komponente** práve vtedy, keď existuje jednofarebná cesta, ktorá ich spája.

Bratislavský kôň vám povedal ako vyzerá graf (jeho vrcholy a hrany), odmietol vám však povedať, akú farbu majú jednotlivé vrcholy. Aby vás pustil do hradu, musíte farby vrcholov zistiť využitím niekoľkých **prefarbovacích experimentov**.

V rámci prefarbovacieho experimentu môžete zmeniť farbu ľubovoľnému počtu vrcholov. Konkrétne, pre vykonanie experimentu musíte vytvoriť pole E dĺžky N , kde hodnota každého prvku $E[i]$ je celé číslo medzi -1 a N (vrátane). Následne, farba vrcholu i (pre $0 \leq i < N$) bude $S[i]$, kde $S[i]$ je:

- $C[i]$, teda originálna farba vrcholu i , ak $E[i] = -1$
- $E[i]$, v opačnom prípade.

Všimnite si, že pri experimentoch môžete používať aj konškrú farbu.

Po každom prefarbovanom experimente vám kôň povie počet jednofarebných komponentov v prefarbenom grafe.

Prefarbenie vrcholov je platné iba počas konkrétneho experimentu, takže **farby vrcholov sa po experimente vrátia na pôvodné hodnoty $C[i]$** .

Vašou úlohou, povedal kôň, je určiť farbu všetkých vrcholov grafu využitím najviac 2 750 prefarbovacích experimentov. Je vám však ochotný dať aj čiastkové body (bránu hradu napoly pootvorí) za to, ak pre každú hranu aspoň správne určíte, či sú na jej koncoch vrcholy rovnakej alebo rozdielnej farby.

Implementačné detaily

Vašou úlohou je implementovať nasledovnú funkciu:

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

- N : počet vrcholov grafu.
- X, Y : polia dĺžky M popisujúce hrany grafu.
- Výstupom tejto funkcie by malo byť pole G dĺžky N , ktoré reprezentuje farby vrcholov grafu.
- Táto funkcia je zavolaná práve raz pre každý vstup.

Vyššie uvedená funkcia môže na vykonanie prefarbovacích experimentov volať nasledovnú funkciu:

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

- E : pole dĺžky N určujúce akým spôsobom majú byť prefarbené vrcholy v tomto experimente.
- Táto funkcia vráti počet jednofarebných komponentov v prefarbenom grafe.
- Táto funkcia môže byť v rámci jedného vstupu zavolaná najviac 2 750-krát.

Testovač **nie je adaptívny**, farby vrcholov sú určené ešte pred volaním funkcie `find_colours`.

Obmedzenia

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $0 \leq X[j] < Y[j] < N$ pre všetky j také, že $0 \leq j < M$.

- $X[j] \neq X[k]$ alebo $Y[j] \neq Y[k]$ pre všetky j a k také, že $0 \leq j < k < M$.
- V grafe existuje cesta medzi ľubovoľnou dvojicou vrcholov.
- $0 \leq C[i] < N$ pre všetky i také, že $0 \leq i < N$.

Subtasks

Podúloha	Skóre	Dodatočné obmedzenia
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Graf na vstupe je cesta: $M = N - 1$ a vrcholy j a $j + 1$ sú susedné pre všetky $0 \leq j < M$.
4	21	Graf na vstupe je úplný: $M = \frac{N \cdot (N-1)}{2}$ a všetky dvojice vrcholov sú susedné.
5	36	Bez dodatočných obmedzení.

V každej podúlohe je možné získať čiastkové body ak sa vám podarí o každej hrane správne určiť, či spája vrcholy rovnakej alebo rôznej farby.

Presnejšie, plný počet bodov za podúlohu dostanete vtedy, ak pre všetky vstupy tejto podúlohy je pole G vrátené funkciou `find_colours` rovnaké ako pole C (teda $G[i] = C[i]$ pre všetky i tak, že $0 \leq i < N$).

Inak môžete získať za podúlohu 50% bodov ak bude pre každý vstup tejto podúlohy platiť:

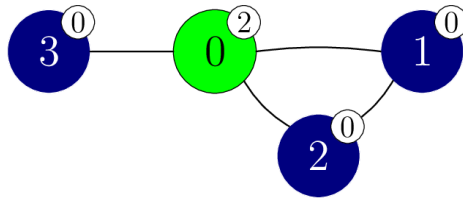
- $0 \leq G[i] < N$ pre všetky i také, že $0 \leq i < N$;
- Pre všetky j také, že $0 \leq j < M$:
 - $G[X[j]] = G[Y[j]]$ práve vtedy keď $C[X[j]] = C[Y[j]]$.

Príklad

Uvažujte nasledovné volanie funkcie:

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

Predpokladajme, že v tomto prípade sú skryté farby vrcholov nasledovné: $C = [2, 0, 0, 0]$, tak ako na obrázku nižšie. (Farby vrcholov na obrázku sú určené poľom C . Hodnoty poľa C navyše nájdete aj v bielych krúžkoch.)



Funkcia môže zavolať `perform_experiment` nasledovne:

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

V tomto volaní nie je prefarbený žiaden vrchol, všetky majú ponechanú pôvodnú farbu.

Vrcholy 1 a 2 majú rovnakú farbu (0) a sú spojené jednofarebnou cestou 1, 2. Preto sa tieto vrcholy nachádzajú v jednom jednofarebnom komponente.

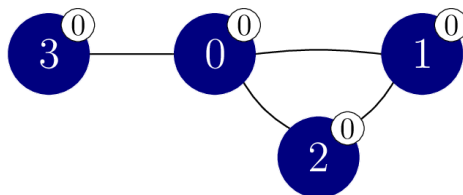
Vrcholy 1 a 3 majú rovnakú farbu (0), nevedie však medzi nimi žiadna jednofarebná cesta. Preto musia byť v dvoch rôznych jednofarebných komponentov.

Dokopy sú v grafe 3 jednofarebné komponenty: $\{0\}$, $\{1, 2\}$ a $\{3\}$. Funkcia `perform_experiment` preto vráti hodnotu 3.

Nasleduje volanie `perform_experiment`:

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

V tomto volaní je prefarbený iba vrchol 0 a to farbou 0, čoho výsledkom je ofarbenie na nasledovnom obrázku:

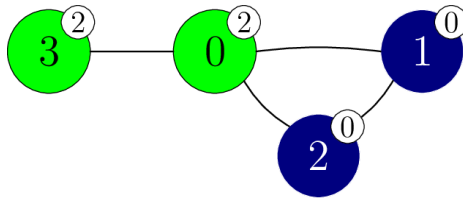


Toto volanie vráti hodnotu 1, keďže všetky vrcholy v takomto farbení patria do jedného jednofarebného komponentu. Z tohto výsledku vieme vydedukovať, že vrcholy 1, 2 a 3 majú farbu 0.

Následne zavoláme:

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

V tomto experimente je prefarbený iba vrchol 3 farbou 2. Výsledné farbenie je na obrázku nižšie. (Všimnite si, že vrchol 0 má už pôvodnú farbu 2.)



Výstupom tohto volania je hodnota 2, keďže graf sa skladá z dvoch jednofarebných komponentov $\{0, 3\}$ a $\{1, 2\}$. Z toho vyplýva, že vrchol 0 má farbu 2.

Funkcia `find_colours` preto vráti pole $[2, 0, 0, 0]$. Keďže toto pole je totožné s C , získame plný počet bodov.

Polovicu bodov by sme mohli získať, ak by funkcia `find_colours` vrátila napríklad pole $[1, 2, 2, 2]$ alebo $[1, 2, 2, 3]$.

Ukázkový testovač

Formát vstupu:

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

Formát výstupu:

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

Hodnota L je dĺžka poľa G , ktoré vrátila funkcia `find_colours` a Q je počet volaní funkcie `perform_experiment`.