

## Sfinks jumbog'i

Buyuk Sfinks sizga jumboq berdi.  $N$  ta tugundan iborat graf berilgan. Tugunlar 0 dan  $N - 1$  gacha raqamlangan. Grafda  $M$  ta 0 dan  $M - 1$  gacha raqamlangan qirralar bor. Har bir qirra ikki turli tugunni bog'laydi va ikki tomonlamali. Aniqrog'i, 0 dan  $M - 1$  gacha bo'lgan har bir  $j$  uchun,  $j$  qirra  $X[j]$  va  $Y[j]$  tugunlarni bog'laydi. Ikkita turli tugun ko'pi bilan bitta qirra orqali bog'langan. Agar ikkita tugun qirra orqali bog'langan bo'lsa, bu ikki tugun **qo'shni** deyiladi.

$v_0, v_1, \dots, v_k$  ( $k \geq 0$  uchun) tugunlar ketma-ketligi **yo'l** bo'lishi uchun, har bir ketma ket  $v_l$  va  $v_{l+1}$  (har bir  $0 \leq l < k$  bo'lgan  $l$  uchun) tugun qo'shni bo'lishi kerak. Shunda  $v_0, v_1, \dots, v_k$  yo'l  $v_0$  va  $v_k$  tugunlarni **bog'laydi** deya olamiz. Berilgan grafda har bir tugun juftligi qandaydir yo'l orqali bog'langan.

0 dan  $N$  gacha raqamlangan  $N + 1$  ta rang mavjud.  $N$  raqamli rang maxsus bo'lib, u **Sfinks rangi** deb ataladi. Har bir tugunni rangi mavjud.  $i$  ( $0 \leq i < N$ ) tugunni rangi  $C[i]$  orqali ifodalanadi. Bir nechta tugunlarni rangi bir xil bo'lishi va ba'zi ranglar hech qanday tugunga belgilanmagan bo'lishi mumkin. Birorta ham tugunni rangi Sfinks rangi emas, ya'ni  $0 \leq C[i] < N$  ( $0 \leq i < N$ ).

$v_0, v_1, \dots, v_k$  (for  $k \geq 0$ ) yo'l **monoxromatik** bo'lishi uchun uning har bir tugunlari rangi bir xil bo'lishi lozim, ya'ni  $C[v_l] = C[v_{l+1}]$  (har bir  $0 \leq l < k$  bo'lgan  $l$  uchun). Shuningdek,  $p$  va  $q$  ( $0 \leq p < N$ ,  $0 \leq q < N$ ) tugunlar bir xil **monoxromatik komponentda** bo'lishi uchun ular monoxromatik yo'l orqali bog'langan bo'lishi shart.

Siz tugunlarni va qirralarni bilasiz, ammo har bir tugun qanday rangda ekanini bilmaysiz. Tugunlarni rangini **qayta bo'yash tajribalari** orqali aniqlamoqchisiz.

Qayta bo'yash tajribalarida, ixtiyoriy ko'p tugunlarni bo'yashingiz mumkin. Aniqroq aytganda, qayta bo'yash tajribasini amalga oshirish uchun, uzunligi  $N$  bo'lgan  $E$  massivni tanlaysiz. Bunda har bir  $i$  ( $0 \leq i < N$ ) uchun  $E[i]$  ning qiymati  $-1$  dan  $N$  gacha butun son bo'lishi mumkin. So'ng, har bir  $i$  tugunning rangi  $S[i]$  ga o'zgaradi,  $S[i]$  ni qiymati esa quyidagiga teng bo'ladi:

- $C[i]$ ga(ya'ni  $i$  tugunning original rangi), agar  $E[i] = -1$ , yoki
- $E[i]$ ga, aks holda.

Bu degani siz qayta bo'yash jarayonida Sfinks rangidan foydalanishingiz mumkin.

Oxiri, har bir  $i$  tugunning rangini  $S[i]$  ga qo'ygandan so'ng, Buyuk Sfinks sizga grafdagı monoxromatik komponentlar sonini aytadi. Yangi ranglar faqat ma'lum bir qayta bo'yash tajribasiga ta'sir qiladi, ya'ni **barcha tugunlar rangi tajribadan so'ng dastlabki holiga qaytadi**.

Sizning vazifangiz, grafdagi tugunlar rangini ko'pi bilan 2750 ta qayta bo'yash tajribalaridan foydalangan holda aniqlashdan iborat. Shuningdek, agar har bir qo'shni tugunlar uchun, ularning rangi bir xil yoki yo'qligini aniqlasangiz, qism ballar olishingiz mumkin.

## Kod yozish detallari

Quyidagi funksiyani kodlashingiz lozim.

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

- $N$ : grafdagi tugunlar soni.
- $X, Y$ : qirralarni ifodalovchi uzunligi  $M$  bo'lgan massivlar.
- Bu funksiya grafdagi tugunlar rangini ifodalovchi uzunligi  $N$  bo'lgan  $G$  massivni qaytarishi lozim.
- Bu funksiya har bir test uchun bir marta chaqiriladi.

Yuqoridagi funksiya qayta bo'yash tajribalarini amalga oshirish uchun quyidagi funksiyani chaqirishi mumkin:

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

- $E$ : graf tugunlari qay tartibda qayta bo'yalishini ifodalovchi uzunligi  $N$  bo'lgan massiv.
- Bu funksiya graf tugunlari  $E$  da berilgan tartib bo'yicha bo'yalganidan keyingi grafdagi monoxromatik komponentlar sonini qaytaradi.
- Bu funksiya ko'pi bilan 2750 marta chaqirilishi mumkin.

Grader **adaptive emas**, ya'ni tugunlarning rangi `find_colours` funksiyasiga chaqiruvlardan oldin belgilab qo'yilgan.

## Chegaralar

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N - 1)}{2}$
- $0 \leq X[j] < Y[j] < N$  har bir  $0 \leq j < M$  bo'lgan  $j$  uchun.
- $X[j] \neq X[k]$  yoki  $Y[j] \neq Y[k]$  har bir  $0 \leq j < k < M$  bo'lgan  $j$  va  $k$  uchun.
- Har bir tugunlar juftligi qandaydir yo'l orqali bog'langan.
- $0 \leq C[i] < N$  har bir  $0 \leq i < N$  bo'lgan  $i$  uchun.

## Qism masalalar

Qism masala	Ball	Qo'shimcha cheklovlar
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Graf bitta yo'ldan iborat : $M = N - 1$ , shuningdek, $j$ va $j + 1$ tugunlar qo'shni ( $0 \leq j < M$ ).
4	21	Graf to'liq: $M = \frac{N \cdot (N-1)}{2}$ va ixtiyoriy ikki tugun bog'langan.
5	36	Qo'shimcha cheklovlar yo'q.

Har bir qism masala uchun, agar dasturingiz har bir tugun uchun ularning rangi bir xil yoki yo'qligini aniqlasa, qism ball olishingiz mumkin.

Aniqroq aytganda, qism masalaning har bir testi uchun `find_colours` tomonidan qaytarilgan  $G$  massiv berilgan  $C$  massiv bilan bir xil bo'lsa (ya'ni har bir  $0 \leq i < N$  bo'lgan  $i$  uchun  $G[i] = C[i]$  bo'lsa), shu qism massiv uchun to'liq ball olasiz. Aks holda, har bir testda agar quyidagi shartlar bajarilsa, qism masalaning 50% ballini olasiz:

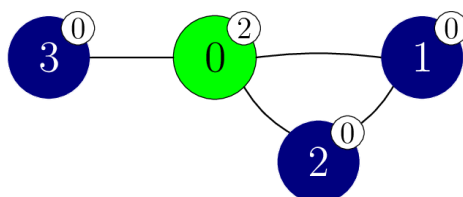
- $0 \leq G[i] < N$  har bir  $0 \leq i < N$  bo'lgan  $i$  uchun;
- Har bir  $0 \leq j < M$  bo'lgan  $j$  uchun:
  - agar  $C[X[j]] = C[Y[j]]$  bo'lsa,  $G[X[j]] = G[Y[j]]$  bo'lishi shart.

## Misol

Quyidagi funksiya chaqiruvini ko'raylik.

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

Bu misol uchun, aytaylik tugunlarning yashirin rangi  $C = [2, 0, 0, 0]$  bo'lsin. Bu graf quyidagi rasmda keltirilgan. Ranglar qo'shimcha ravishda tugunlarning yonida sonlar orqali ham keltirilgan.



Bu funksiya `perform_experiment`ni quyidagicha chaqirishi mumkin.

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

Bu chaqiruvda, hech qaysi tugun qayta bo'yalmadi, chunki hamma tugunlarni rangi o'z holicha qoldi.

1- va 2- tugunlarni ko'raylik. Ularning ikkovini ham rangi 0 va 1, 2 yo'l monoxromatik yo'l. Natijada, 1 va 2 tugunlar bitta monoxromatik komponentada.

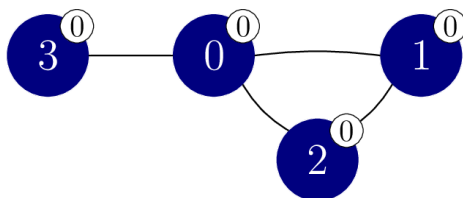
1- va 3- tugunlarni ko'raylik. Ularning rangi bir xilda 0 bo'lsada, ularni o'rtasida monoxromatik yo'l bo'lmagani tufayli, ular turli xil monoxromatik komponentalarda joylashgan.

Umumiy olganda, 3 ta monoxromatik komponentalar mavjud:  $\{0\}$ ,  $\{1, 2\}$ , va  $\{3\}$ . Shuning uchun, funksiya 3 qaytaradi.

Ana endi, funksiya `perform_experiment`ni quyidagicha chaqirishi mumkin.

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

Bu chaqiruvda, faqat 0-tugun 0 rangiga qaytadan bo'yaldi, natijada graf quyidagi holga keladi.

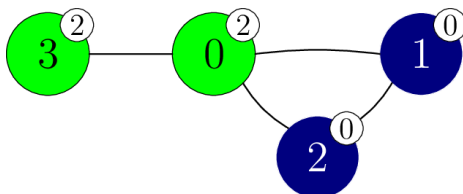


Bunda funksiya 1 qaytaradi, chunki barcha tugunlar bitta monoxromatik komponentada. Bu orqali, 1, 2 va 3-tugunlarning rangi 0 ekanini aniqlab olishimiz mumkin.

So'ng, funksiya `perform_experiment`ni quyidagicha chaqirishi mumkin.

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

Bu chaqiruvda 3-tugun 2 rangiga qaytadan bo'yaldi, natijada graf quyidagi holga keladi.



Bu funksiya 2 qaytaradi, chunki grafda  $\{0, 3\}$  va  $\{1, 2\}$  monoxromatik komponentalar mavjud. Bundan 0-tugunning rangi 2 ekanini aniqlab olish mumkin.

`find_colours` funksiyasi so'ng  $[2, 0, 0, 0]$  massivni qaytaradi.  $C = [2, 0, 0, 0]$  bo'lgani tufayli, to'liq ball beriladi.

50% ballni olish uchun, turli xil javoblar mavjud, masalan  $[1, 2, 2, 2]$  yoki  $[1, 2, 2, 3]$ .

# Namunaviy Grader

Kiritish formati:

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

Chiqarish formati:

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

Bu yerda,  $L$  `find_colours` tomonidan qaytarilgan  $G$  massivni uzunligini,  $Q$  esa `perform_experiment` ga chaqiruvlar sonini qaytaradi.